Intel® 82576 Gigabit Ethernet Controller Datasheet

LAN Access Division (LAD)

PRODUCT FEATURES

External Interfaces

- PCIe* v2.0 (2.5 GT/s) x4/x2/x1; called PCIe in this document
- MDI (Copper) standard IEEE 802.3 Ethernet interface for 1000BASE-T, 100BASE-TX, and 10BASE-T applications (802.3, 802.3u, and 802.3ab)
- Serializer-Deserializer (SERDES) to support 1000Base-SX/BX/LX (optical fiber) for Gigabit backplane applications.
- SGMII for SFP/external PHY connections
- NC-SI or SMBus for Manageability connection to BMC.
- IEEE 1149.1 JTAG

Intel® I/O Acceleration Technology

- Stateless offloads (Header split, RSS)
- Direct Cache Access
- Receive side Coalescing (RSC)

Virtualization Ready

- Next Generation VMDq support (8 VMs)
- PCI-SIG Single Root I/O Virtualization (Direct assignment)
- Queues per port: 16 TX queues and 16 RX queues

Full-Spectrum Security

- IPsec (256 SA’s)

Additional Product Details

- 25mm x 25mm Package
- Power 2.4W (max)
- Support for PCI 3.0 Vital Product Data
- Full Data Path ECC Protection
- IPMI BMC Pass-thru; Multi-drop NC-SI
- IEEE 1588 Precision Time Protocol Support.
- Layout Compatible with 82575.

Revision 2.0
July 2008

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

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

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

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

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

Hyper-Threading Technology requires a computer system with an Intel® Pentium® 4 processor supporting HT Technology and a HT Technology enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. See http://www.intel.com/products/ht/Hyperthreading_more.htm for additional information.

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

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

Intel and Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Revision History

Date	Revision	Description
June 2007	0.5	Initial availability.
November 2007	1.0	Updates and corrections.
May 2008	1.9	PRQ release.[1]
July 2009	2.0	SRA release. After SRA, changes that impact function are detailed in this table.

[1] The assigned version number (1.9) is the designation defined for PRQ support. Numbers in the series have been skipped without being assigned to actual releases.

Contents

1. Introduction

The Intel® 82576 GbE Controller is a single, compact, low power component that offers two fully-integrated Gigabit Ethernet Media Access Control (MAC) and physical layer (PHY) ports. This device uses the PCIe v2.0 (2.5GT/s). The 82576 enables two-port implementation in a relatively small area and can be used for server system configurations such as rack mounted or pedestal servers, where the 82576 can be used as add-on NIC or LAN on Motherboard (LOM) design. Another system configuration is blade servers, where it can be used as LOM. The 82576 can also be used in embedded applications such as switch add-on cards and network appliances.

1.1 Scope

This document presents the external architecture (including device operation, pin descriptions, register definitions, etc.) for the 82576, a dual 10/100/1000 LAN controller.

This document is intended to be a reference for software device driver developers, board designers, test engineers, or others who may need specific technical or programming information.

1.2 Terminology and Acronyms

Table 1-1. Glossary

Definition	Meaning
1000BASE-BX	1000BASE-BX is the PICMG 3.1 electrical specification for transmission of 1 Gb/s Ethernet or 1 Gb/s fibre channel encoded data over the backplane.
1000BASE-CX	1000BASE-X over specialty shielded 150 Ω balanced copper jumper cable assemblies as specified in IEEE 802.3 Clause 39.
1000BASE-T	1000BASE-T is the specification for 1 Gb/s Ethernet over category 5e twisted pair cables as defined in IEEE 802.3 clause 40.
AH	IP Authentication Header - An IPsec header providing authentication capabilities defined in RFC 4302.
b/w	Bandwidth.
BIOS	Basic Input/Output System.
BMC	Baseboard Management Controller.
BT	Byte Time.
BWG	Bandwidth Group.
CA	Secure Connectivity Association (CA): A security relationship, established and maintained by key agreement protocols.
CPID	Congestion Point Identifier.
DCA	Intel® QuickData (Direct Cache Access).
DFP	Deficit Fixed Priority.
DFT	Design for Testability.

Table 1-1. Glossary (Continued)

Definition	Meaning
DQ	Descriptor Queue.
EEPROM	Electrically Erasable Programmable Memory. A non-volatile memory located on the LAN controller that is directly accessible from the host.
EOP	End of Packet.
ESP	IP Encapsulating Security Payload - An IPsec header providing encryption and authentication capabilities defined in RFC 4303.
FC	Flow Control.
Firmware (FW)	Embedded code on the LAN controller that is responsible for the implementation of the NC-SI protocol and pass through functionality.
Host Interface	RAM on the LAN controller that is shared between the firmware and the host. RAM is used to pass commands from the host to firmware and responses from the firmware to the host.
HPC	High - Performance Computing.
IPC	Inter Processor Communication.
IPG	Inter Packet Gap.
LAN (auxiliary Power-Up)	The event of connecting the LAN controller to a power source (occurs even before system power-up).
LOM	LAN on Motherboard.
LSO	Large Send Offload.
MAC	Media Access Control.
MDIO	Management Data Input/Output Interface over MDC/MDIO lines.
MIFS/MIPG	Minimum Inter Frame Spacing/Minimum Inter Packet Gap.
MMW	Maximum Memory Window.
MSS	Maximum Segment Size.
NIC	Network Interface Controller.
PCS	Physical Coding Sub layer.
PF	Physical Function (in a virtualization context).
PHY	Physical Layer Device.
PMA	Physical Medium Attachment.
PMD	Physical Medium Dependent.
NC-SI	Reduced Media Independent Interface (Reduced MII).
SA	Source Address.
SCI	A globally unique identifier for a secure channel, comprising a globally unique MAC Address and a Port Identifier, unique within the system allocated that address.
SDP	Software Defined Pins.
SerDes	Serializer and De-Serializer Circuit.
SFD	Start Frame Delimiter.
SGMII	Serialized Gigabit Media Independent Interface.
SMBus	System Management Bus. A bus that carries various manageability components, including the LAN controller, BIOS, sensors and remote-control devices.
TSO	Transmit Segmentation offload - A mode in which a large TCP/UDP I/O is handled to the device and the device segments it to L2 packets according to the requested MSS.

Table 1-1. Glossary (Continued)

Definition	Meaning
VF	Virtual Function.
VM	Virtual Machine.
VPD	Vital Product Data (PCI protocol).

1.2.1 External Specification and Documents

The 82576 implements features from the following specifications.

1.2.1.1 Network Interface Documents

1. IEEE standard 802.3, 2005 Edition (Ethernet). Incorporates various IEEE Standards previously published separately. Institute of Electrical and Electronic Engineers (IEEE).
2. IEEE standard 1149.1, 2001 Edition (JTAG). Institute of Electrical and Electronics Engineers (IEEE)
3. IEEE standard 802.1Q for VLAN
4. PICMG3.1 Ethernet/Fibre Channel Over PICMG 3.0 Draft Specification January 14, 2003 Version D1.0
5. Serial-GMII Specification, Cisco Systems document ENG-46158, Revision 1.7.
6. INF-8074i Specification for SFP (Small Formfactor Pluggable) Transceiver (ftp://ftp.seagate.com/sff)

1.2.1.2 Host Interface documents

1. PCI-Express 2.0 Base specification, Revision 1.0
2. PCI Specification, version 3.0
3. PCI Bus Power Management Interface Specification, Rev. 1.2, March 2004
4. Advanced Configuration and Power Interface Specification, Rev 2.0b, October 2002

1.2.1.3 Virtualization documents

1. PCI-Express Single Root I/O Virtualization and Sharing Specification rev 0.9
2. PCI sig Alternative Routing-ID Interpretation (ARI) ECN (http://teamsites.ch.ith.intel.com/sites/PASDPA/PCIe/PCI%20Express%20Product_Spec%20Coordination/pages/PCISIG%20WIP%20Docs.aspx)

1.2.1.4 Networking Protocol documents

1. IPv4 specification (RFC 791)
2. IPv6 specification (RFC 2460)
3. TCP/UDP specification (RFC 793/768)
4. SCTP specification (RFC 2960)
5. ARP specification (RFC 826)
6. EUI-64 specification, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.

1.2.1.5 Manageability documents

1. DMTF Network Controller Sideband Interface (NC-SI) Specification rev 0.7
2. System Management Bus (SMBus) Specification, SBS Implementers Forum, Ver. 2.0, August 2000

1.2.1.6 Security documents

1. IEEE P802.1AE/D5.1 — Draft Standard for Local and Metropolitan Area Networks — Media Access Control (MAC) Security.
2. The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) (RFC 4106)
3. IP Authentication Header (AH) (RFC 4302)
4. IP Encapsulating Security Payload (ESP) (RFC 4303)
5. The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH (RFC 4543).

1.2.2 Intel Application Notes

1. Intel® Ethernet Controllers Loopback Modes - application note.

1.3 Product Overview

The 82576 supports 2 ports with either an internal PHY or a SerDes or SGMII port which may connected to an external PHY or directly to a blade connection for MAC to MAC communication.

1.3.1 System Configurations

The 82576 targets server system configurations such as rack mounted or pedestal servers, where the 82576 can be used as add-on NIC or LAN on Motherboard (LOM) design. Another system configuration is blade servers, where it can be used as LOM. The 82576 can also be used in embedded applications such as switch add-on cards and network appliances.

1.4 External Interface

1.4.1 PCIe* Interface

The PCIe v2.0 (2.5GT/s) interface is used by the 82576 as a host interface. It supports x4, x2 and x1 configurations, while each lane runs at 2.5 GHz speed. The maximum aggregated raw bandwidth for a typical x4 configuration is 8 Gb/s in each direction. See Section 2.1.1 for a full pin description. The timing characteristics of this interface are defined in PCI Express Card Electromechanical Specification rev 1.0 and in the PCIe v2.0 (2.5GT/s) specification.

1.4.2 Network interfaces

Two independent interfaces are used to connect the two 82576 ports to external devices. The following protocols are supported:

- 10BASE-T and 100BASE-T.
- 1000Base-T interface to attach directly to a CAT 5e wire.
- SerDes interface to connect over a backplane to another SerDes compliant device or to an optic module.
- SGMII interface to attach to an external PHY, either on board or via an SFP module. The SGMII shares the same interface as the SerDes.
- MDI (Copper) support for standard IEEE 802.3 Ethernet interface for 1000BASE-T, 100BASE-TX, and 10BASE-T applications (802.3, 802.3u, and 802.3ab).

See Section 2.1.8.2 and Section 2.1.6 for full pin description; Section 11.4.4.1 to Section 11.4.4.3 for timing characteristics of this interface.

1.4.3 EEPROM Interface

The 82576 uses an EEPROM device for storing product configuration information. Several words of the EEPROM are accessed automatically by the 82576 after reset in order to provide pre-boot configuration data that must be available to the 82576 before it is accessed by host software. The remainder of the stored information is accessed by various software modules used to report product configuration, serial number, etc.

The 82576 is intended for use with an SPI (4-wire) serial EEPROM device such as an AT25040AN or compatible. See Section 2.1.2 for full pin description and Section 11.4.3.5 for timing characteristics of this interface.

The 82576 also supports an EEPROM-less mode, where all of the setup is done by software.

1.4.4 Serial Flash Interface

The 82576 provides an external SPI serial interface to a Flash or Boot ROM device such as the Atmel* AT25F1024 or AT25FB512. The 82576 supports serial Flash devices with up to 64 Mb (8 MB) of memory. The size of the Flash used by the 82576 can be configured by the EEPROM. See Section 2.1.2 for full pin description and Section 11.4.3.4 for timing characteristics of this interface.

Note: Though the 82576 supports devices with up to 8 MB of memory, bigger devices can also be used. Accesses to memory beyond the Flash device size results in access wrapping as only the lower address bits are used by the Flash device.

1.4.5 SMBus Interface

SMBus is an optional interface for pass-through and/or configuration traffic between a BMC and the 82576.

The 82576's SMBus interface can be configured to support both slow and fast timing modes. See Section 2.1.3 for full pin description and Section 11.4.3.3 for timing characteristics of this interface.

1.4.6 NC-SI Interface

NC-SI and SMBus interfaces are optional for pass-through and/or configuration traffic between a BMC and the 82576. The NC-SI interface meets the DMTF NC-SI Specification, Rev. 1.0.0.a.

1.4.7 MDIO/2 wires Interfaces

The 82576 implements two management Interfaces for control of an optional external PHY. Each interface can be either a 2 wires interface used to control an SFP module or MDIO/MDC Management Interface for control plane connection between the MAC and PHY devices (master side). This interface provides the MAC and software with the ability to monitor and control the state of the PHY. The 82576 supports the data formats of 802.3 clause 22. Each MDIO interface should be connected to the relevant PHY.

See Section 2.1.7 for full pin description and Section 11.4.3.9 for timing characteristics of this interface.

1.4.8 Software-Definable Pins (SDP) Interface (General-Purpose I/O)

The 82576 has four software-defined pins (SDP pins) per port that can be used for miscellaneous hardware or software-control purposes. These pins can be individually configurable to act as either input or output pins. The default direction of each pin is configurable via the EEPROM (see Section 6.2.8

and Section 6.2.9), as well as the default value of all pins configured as outputs. To avoid signal contention, all pins are set as input pins until the EEPROM configuration is loaded. All four of the SDP pins can be configured for use as general-purpose interrupt (GPI) inputs. To act as GPI pins, the desired pins must be configured as inputs. A corresponding GPI interrupt-detection enable bit is then used to enable rising-edge detection of the input pin (rising-edge detection occurs by comparing values sampled at the internal clock rate, as opposed to an edge-detection circuit). When detected, a corresponding GPI interrupt is indicated in the Interrupt Cause register.

The use, direction, and values of SDP pins are controlled and accessed using fields in the Device Control (CTRL) register and Extended Device Control (CTRL_EXT) register.

See Section 2.1.5 for full pin description of this interface.

1.4.9 LEDs Interface

The 82576 implements four output drivers per port intended for driving external LED circuits. Each of the four 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. Furthermore, the hardware-default configuration for all LED outputs can be specified via EEPROM fields (see Section 6.2.17 and Section 6.2.19), thereby supporting LED displays configurable to a particular OEM preference.

See Section 2.1.8.1 for full pin description of this interface.

See Section 7.5 for more detailed description of LED behavior.

1.5 Comparing Product Features

The following tables compare features of similar Intel components.

Table 1-2. 82576 Features

Feature	82576	82575	82571EB
Number of ports	2	2	2
Serial FLASH interface	Y	Y	Y
4-wire SPI EEPROM interface	Y	Y	Y
Configurable LED operation for software or OEM custom-tailoring of LED displays	Y	Y	Y
Protected EEPROM space for private configuration	Y	Y	Y
Device Disable capability	Y	Y	Y
Package size (mm x mm)	25x25	25x25	17x17
Watchdog timer	Y	Y	N

Table 1-3. 82576 Network Features

Feature	82576	82575	82571EB
Half duplex at 10/100 Mb/s operation and full duplex operation at all supported speeds	Y	Y	Y
10/100/1000 Copper PHY integrated on-chip	Y	Y	Y
Jumbo frames supported	Y	Y	Y

Table 1-3. 82576 Network Features (Continued)

Size of jumbo frames supported	9.5 KB	9.5 KB	9 KB
Flow control support: send/receive PAUSE frames and receive FIFO thresholds	Y	Y	Y
Statistics for management and RMON	Y	Y	Y
802.1q VLAN support	Y	Y	Y
SerDes interface for external PHY connection or system interconnect	Y	Y	Y
SGMII interface for embedded applications	Y	Y	N
Fiber/copper auto-sense*	Y	Y	N
SerDes support of non-Auto-Negotiation partner	Y	Y	N
SerDes signal detect	Y	Y	N

Table 1-4. 82576 Host Interface Features

Feature	82576	82575	82571EB
PCIe revision	2.0	2.0	1.0a
PCIe physical layer	(2.5 GT/s)	2.5 GT/s)	2.5 GT/s)
Bus width	x1, x2, x4	x1, x2, x4	x1, x2, x4
64-bit address support for systems using more than 4 GB of physical memory	Y	Y	Y
Outstanding requests for Tx buffers	4	4	4
Outstanding requests for Tx descriptors	1	1	1
Outstanding requests for Rx descriptors	1	1	1
Credits for posted writes	2	2	2
Max payload size supported	512 B	256 B	256 B
Max request size supported	512 B	512 B	256 B
Link layer retry buffer size	2 KB	2 KB	2 KB
Vital Product Data (VPD)	Y	N	N

Table 1-5. 82576 LAN Functions Features

Feature	82576	82575	82571EB
Programmable host memory receive buffers	Y	Y	Y
Descriptor ring management hardware for transmit and receive	Y	Y	Y
ACPI register set and power down functionality supporting D0 & D3 states	Y	Y	Y
Software controlled global reset bit (resets everything except the configuration registers)	Y	Y	Y
Software Definable Pins (SDP) - per port	4	4	4
Four SDP pins can be configured as general purpose interrupts	Y	Y	Only 2
Wake up	Y	Y	Y
IPv6 wake-up filters	Y	Y	Y
Configurable (through the EEPROM) flexible filter	Y	Y	Y

Table 1-5. 82576 LAN Functions Features (Continued)

Default configuration by the EEPROM for all LEDs for pre-driver functionality	Y	Y	Y
LAN function disable capability	Y	Y	Y
Programmable memory transmit buffers (up to 32 KB)	Y	Y	Y
Double VLAN	Y	Y	N
IEEE 1588	Y	N	N

Table 1-6. 82576 LAN Performance Features

Feature	82576	82575	82571EB
TCP segmentation offload Up to 256 KB	Y	Y	Y
IPv6 support for IP/TCP and IP/UDP receive checksum offload	Y	Y	Y
Fragmented UDP checksum offload for packet reassembly	Y	Y	Y
Message Signaled Interrupts (MSI)	Y	Y	Y
Message Signaled Interrupts (MSI-X)	Y	Y	N
Packet interrupt coalescing timers (packet timers) and absolute-delay interrupt timers for both transmit and receive operation	Y	N	N
Interrupt throttling control to limit maximum interrupt rate and improve CPU utilization	Y	Y	Y
Rx packet split header	Y	Y	Y
Receive Side Scaling (RSS) number of queues	Up to 16	4	2
Total number of RX queues per port	16	4	2
Total number of TX queues per port	16	4	2
RX header replication Low latency interrupt DCA support TCP timer interrupts No snoop Relax ordering	Yes to all	Yes to all	Y N N N Y Y
TSO interleaving for reduced latency	Y	N	N
Receive side coalescing	N	N	N
SCTP receive and transmit checksum offload	Y	N	N
UDP TSO	Y	N	N

Table 1-7. 82576 Virtualization Features

Feature	82576	82575	82571EB
Support for Virtual Machines Device queues (VMDq)	8 pools	4	N
PCI-SIG SR IOV	8 VF	N	N
Multicast/Broadcast Packet replication	Y	N	N
VM to VM Packet forwarding	Y	N	N
Traffic shaping	Y	N	N
MAC addresses	24	16	15

Table 1-7. 82576 Virtualization Features (Continued)

MAC and VLAN anti-spoofing	Y	N	N
VLAN filtering	Per pool	Global	Global
Per-pool statistics	Y	N	N
Per-pool off loads	Y	Partial	N
Per-pool jumbo support	Y	N	N
Mirroring rules	4	0	0

Table 1-8. 82576 Manageability Features

Feature	82576	82575	82571EB
Advanced pass-through-compatible management packet transmit/receive support	Y	Y	Y
Manageability support for ASF 1.0 and Alert on LAN 2.0	N	Y	Y
SMBus interface to external BMC	Y	Y	Y
DMTF NC-SI protocol standards support	Y	Y	N
L2 address filters	4	4	1
VLAN L2 filters	8	8	4
Flex L3 port filters	16	16	3
Flex TCO filters	4	4	2
L3 address filters (IPv4)	4	4	4
L3 address filters (IPv6)	4	4	1
SOL/IDE/KCS functions	N	N	Y
Fast Management Bus support	N	N	Y

Table 1-9. 82576 Security Features

Feature	82576	82575	82571EB
Integrated IPSec Offload Engine • Security Associations - Rx • Security Associations - Tx • IP Authentication Header (AH) protocol • IP Encapsulating Security Payload (ESP) for authentication and/or Encryption. • AES-128-GMAC (128-bit key) engine • IPv4 and IPv6 support (without options or extensions)	Y 256 256 Y Y Y Y	N	N

1.6 Overview of New Capabilities

The following section describes features added in Intel® 82576 GbE Controller that are new related to 82575.

1.6.1 IPsec Off Load for Flows

The 82576 supports IPsec off load for a given number of flows. It is the operating system's responsibility to submit to hardware the most loaded flows, in order to take maximum benefits of the IPsec off-load in terms of CPU utilization savings. Main features are:

- Off-load IPsec for up to 256 Security Associations (SA) for each of Tx and Rx.
- AH and ESP protocols for authentication and encryption

AES-128-GMAC and AES-128-GCM crypto engines:

- Transport mode encapsulation
- IPv4 and IPv6 versions (no options or extension headers)

1.6.2 Performance

The 82576 provides improves the following performance vectors:

- Latency - The 82576 reduces end-to-end latency for high priority traffic in presence of other traffic. Specifically, the 82576 reduces the delay caused by preceding TSO packets and can be interleaved with other packets going to the wire. Interleaving is done at the Ethernet packet boundary, therefore reducing the maximum delay due to a TSO from a TSO-amount of data to an MTU-amount of data.
- CPU Utilization - The 82576 supports reducing CPU utilization in a virtualized system by incorporating enhancements to the VMDq feature.

1.6.2.1 Tx Descriptor Write-Back

This functionality is an improvement to the way Tx descriptors are written back to memory. Instead of writing back the *DD* bit into the descriptor location, the head pointer is updated in system memory. The head pointer is updated based on the *RS* bit or prior to expiration of the corresponding interrupt vector.

1.6.3 Rx and Tx Queues

The number of Tx and Rx queues in the 82576 was increased to 16 queues.

1.6.4 Interrupts

The following changes in the interrupt scheme are implemented in the 82576:

- Rate controlling of low latency interrupts
- Extensions to the low latency interrupt filters to enable immediate interrupt by full 5-tuple matching

1.6.5 Virtualization

1.6.5.1 PCI SR IOV

The 82576supports the PCI-SIG Single-Root I/O Virtualization and Sharing specification (SR-IOV), including the following functionality:

- Support for up to 8 virtual functions.
- Partial replication of PCI configuration space
- Allocation of MMIO space per virtual function
- Allocation of a requester ID per virtual function

- Virtualization of interrupts

1.6.5.2 Packets Classification

Received unicast packets are forwarded to the appropriate VM queue based on their unicast L2 address. Broadcast and Multicast (MC) packets, however, might need to be forwarded to multiple VMs. Multicast is commonly used to share information among a group of systems.

Received MC packets are forwarded to their destination VMs based on mapping between the MC address and the target VMs.

Broadcast packets that are VLAN tagged are forwarded to destination VMs based on their VLAN tag. Note that a VM might be associated with multiple VLAN addresses. A broadcast packet that is not VLAN tagged can be optionally forwarded to all VMs.

Packet forwarding services inter-VM communication by forwarding transmit packets from a transmit queue to an Rx software queue. The motivation to execute packet forwarding in the 82576 is in direct assignment architecture, where it is desired that a guest VM interacts directly with the 82576 using a standard device driver. If packet forwarding is to be done by system software, the guest VM (its device driver) needs to filter local packets and forward those to a software switch to forward.

Transmit packets with a local destination are classified based on the same criteria as packets received from the wire.

1.6.5.3 Hardware Virtualization

This section covers replication of hardware resources beyond the scope of PCI resources handled by PCI SR-IOV. The following features are supported:

- Interrupts – part of the interrupts are assigned per VM.
- Statistics – enable read access to VMs in direct assignment model without the clear-on-read side effect.
- Storm control - if an unusually high bandwidth of broadcast or multicast packets is detected, the 82576 can be configured to drop broadcast or multicast packets until the storm condition is over.
- Security features: VLAN and MAC anti-spoof are supported as well as insertion of VLAN according to the physical function control.

1.6.5.4 Bandwidth Allocation

The 82576 allows allocation of transmit bandwidth among the virtual interfaces to avoid unfair use of bandwidth by a single VM.

1.6.6 VPD

The 82576 supports the Vital Product Data (VPD) capability defined in the PCI Specification, version 3.0.

1.6.7 64 bit BARs support

The 82576 supports different configuration of the I/O and MMIO Base Address Registers to allow support of 64 bit mappings of BARs.

1.6.8 IEEE 1588 -Precision Time Protocol (PTP)

The IEEE 1588 International Standard enables networked Ethernet equipment to synchronize internal clocks according to a network master clock. The protocol is implemented in software, with the 82576 providing accurate time measurements of special Tx and Rx packets close to the Ethernet link. These packets measure the latency between the master clock and an end-point clock in both link directions. The endpoint can then acquire an accurate estimate of the master time by compensating for link latency.

The 82576 provides the following support for the IEEE 1588 protocol:

- Detection of specific PTP Rx packets and capturing the time of arrival of such packets in dedicated CSRs
- Detection of specific PTP Tx packets and capturing the time of transmission of such packets in dedicated CSRs
- A software-visible reference clock for the previously mentioned time captures.
- Both the L2 based and the UDP based version of the protocol are supported.
- Generation of an external clock on one of the SDPs.
- Triggering of external devices based on internal clock.
- Timestamps of external events.

1.7 Device Data Flows

1.7.1 Transmit Data Flow

Tx data flow provides a high level description of all data/control transformation steps needed for sending Ethernet packets to the line.

Table 1-10. Transmit Data Flow

Step	Description
1	The host creates a descriptor ring and configures one of the 82576's transmit queues with the address location, length, head and tail pointers of the ring (one of 16 available Tx queues).
2	The host is requested by the TCP/IP stack to transmit a packet, it gets the packet data within one or more data buffers.
3	The host initializes descriptor(s) that point to the data buffer(s) and have additional control parameters that describe the needed hardware functionality. The host places that descriptor in the correct location at the appropriate Tx ring.
4	The host updates the appropriate queue tail pointer (TDT)
5	The 82576's DMA senses a change of a specific TDT and as a result sends a PCIe request to fetch the descriptor(s) from host memory.
6	The descriptor(s) content is received in a PCIe read completion and is written to the appropriate location in the descriptor queue internal cache.
7	The DMA fetches the next descriptor from the internal cache and processes its content. As a result, the DMA sends PCIe requests to fetch the packet data from system memory.
8	The packet data is received from PCIe completions and passes through the transmit DMA that performs all programmed data manipulations (various CPU off loading tasks as checksum off load, TSO off load, etc.) on the packet data on the fly.
9	While the packet is passing through the DMA, it is stored into the transmit FIFO. After the entire packet is stored in the transmit FIFO, it is forwarded to the transmit switch module.
10	If the packet destination is also local, it is sent also to the local switch memory and join the receive path.
11	The transmit switch arbitrates between host and management packets and eventually forwards the packet to the Security engine.

Table 1-10. Transmit Data Flow (Continued)

12	The security engine optionally applies L3 (IPsec) encryption or authentication and forwards the packet to the MAC.
13	The MAC appends the L2 CRC to the packet and sends the packet to the line using a pre-configured interface.
14	When all the PCIe completions for a given packet are done, the DMA updates the appropriate descriptor(s).
15	After enough descriptors are gathered for write back or the interrupt moderation timer expires, the descriptors are written back to host memory using PCIe posted writes. Alternatively, the head pointer can only be written back.
16	After the interrupt moderation timer expires, an interrupt is generated to notify the host device driver that the specific packet has been read to the 82576 and the driver can release the buffers.

1.7.2 Rx Data Flow

Rx data flow provides a high level description of all data/control transformation steps needed for receiving Ethernet packets.

Table 1-11. Receive Data Flow

Step	Description
1	The host creates a descriptor ring and configures one of the 82576's receive queues with the address location, length, head, and tail pointers of the ring (one of 16 available Rx queues).
2	The host initializes descriptors that point to empty data buffers. The host places these descriptors in the correct location at the appropriate Rx ring.
3	The host updates the appropriate queue tail pointer (RDT).
4	The 82576's DMA senses a change of a specific RDT and as a result sends a PCIe request to fetch the descriptors from host memory.
5	The descriptors content is received in a PCIe read completion and is written to the appropriate location in the descriptor queue internal cache.
6	A packet enters the Rx MAC. The RX MAC checks the CRC of the packet.
7	The MAC forwards the packet to an Rx filter
8	If the packet is an IPSec packet and the adequate key is stored in the hardware, the packet is decrypted and authenticated.
9	If the packet matches the pre-programmed criteria of the Rx filtering, it is forwarded to the Rx FIFO. VLAN and CRC are optionally stripped from the packet and L3/L4 checksum are checked and the destination queue is fixed.
10	The receive DMA fetches the next descriptor from the internal cache of the appropriate queue to be used for the next received packet.
11	After the entire packet is placed into the Rx FIFO, the receive DMA posts the packet data to the location indicated by the descriptor through the PCIe interface. If the packet size is greater than the buffer size, more descriptors are fetched and their buffers are used for the received packet.
12	When the packet is placed into host memory, the receive DMA updates all the descriptor(s) that were used by packet data.
12	After enough descriptors are gathered for write back or the interrupt moderation timer expires or the packet requires immediate forwarding, the receive DMA writes back the descriptor content along with status bits that indicate the packet information including what off loads were done on that packet.
13	After the interrupt moderation timer completes or an immediate packet is received, the 82576 initiates an interrupt to the host to indicate that a new received packet is already in host memory.
14	Host reads the packet data and sends it to the TCP/IP stack for further processing. The host releases the associated buffers and descriptors once they are no longer in use.

§ §

NOTE: This page intentionally left blank.

2. Pin Interface

2.1 Pin Assignment

The 82576 is packaged in 25mmx25mm FCBGA package with 1 mm ball pitch.

Table 2-1. Signal Type Definition

Type	Description	DC specification
In	Input is a standard input-only signal.	See Section 11.4.2.2
Out	Totem Pole Output is a standard active driver.	See Section 11.4.2.2
T/S	Tri-State is a bi-directional, tri-state input/output pin.	See Section 11.4.2.2
O/D	Open Drain allows multiple devices to share as a wire-OR.	See Section 11.4.2.3
NC-SI-in	Input signal	See Section 11.4.2.4
NC-SI-out	Output signal	See Section 11.4.2.4
A	Analog PHY signals	See Section 11.4.5
A-in	Analog input signals	See Section 11.4.4
A-out	Analog output signals	See Section 11.4.4
B	Input bias	See Section 11.4.7

2.1.1 PCIe

The AC specification for these pins is described in Chapter 11..

Table 2-2. PCIe Pins

Symbol	Ball #	Type	Name and Function
PE_CLK_p PE_CLK_n	N2 N1	A-in	PCIe* Differential Reference Clock in: A 100MHz differential clock input. This clock is used as the reference clock for the PCIe* Tx/Rx circuitry and by the PCIe* core PLL to generate clocks for the PCIe* core logic.
PET_0_p PET_0_n	D2 D1	A-out	PCIe* Serial Data output: A serial differential output pair running at 2.5Gb/s. This output carries both data and an embedded 2.5GHz clock that is recovered along with data at the receiving end.
PET_1_p PET_1_n	H2 H1	A-out	PCIe* Serial Data output: A serial differential output pair running at 2.5Gb/s. This output carries both data and an embedded 2.5GHz clock that is recovered along with data at the receiving end.
PET_2_p PET_2_n	R2 R1	A-out	PCIe* Serial Data output: A serial differential output pair running at 2.5Gb/s. This output carries both data and an embedded 2.5GHz clock that is recovered along with data at the receiving end.
PET_3_p PET_3_n	W2 W1	A-out	PCIe* Serial Data output: A serial differential output pair running at 2.5Gb/s. This output carries both data and an embedded 2.5GHz clock that is recovered along with data at the receiving end.
PER_0_p PER_0_n	F2 F1	A-in	PCIe* Serial Data input: A Serial differential input pair running at 2.5Gb/s. An embedded clock present in this input is recovered along with the data.

Table 2-2. PCIe Pins (Continued)

PER_1_p PER_1_n	K2 K1	A-in	PCIe* Serial Data input: A Serial differential input pair running at 2.5Gb/s. An embedded clock present in this input is recovered along with the data.
PER_2_p PER_2_n	U2 U1	A-in	PCIe* Serial Data input: A Serial differential input pair running at 2.5Gb/s. An embedded clock present in this input is recovered along with the data.
PER_3_p PER_3_n	AA2 AA1	A-in	PCIe* Serial Data input: A Serial differential input pair running at 2.5Gb/s. An embedded clock present in this input is recovered along with the data.
PE_WAKE_N	AC20	O/D	WAKE: Pulled to ‘0’ to indicate that a Power Management Event (PME) is pending and the PCI Express link should be restored. Defined in the PCI Express specifications.
PE_RST_N	AC9	In	Power and Clock Good Indication: Indicates that power and PCI Express reference clock are within specified values. Defined in the PCI Express specifications.
PE_RCOMP	L1	B	Impedance compensation. Connect to ground through an external 1.4 Kohm 1% 100ppm resistor for impedance compensation. See Figure 11-13 for details.

2.1.2 Flash and EEPROM Ports

The AC specification for these pins is described in Section 11.4.3.4 to Section 11.4.3.5.

Table 2-3. Flash and EEPROM Ports

Symbol	Ball #	Type	Name and Function
FLSH_SI	AC14	T/S	Serial Data output to the Flash
FLSH_SO	AD14	In	Serial Data input from the Flash
FLSH_SCK	AD15	T/S	Flash serial clock Operates at ~20MHz.
FLSH_CE_N	AC15	T/S	Flash chip select Output
EE_DI	A21	T/S	Data output to EEPROM
EE_DO	A20	In	Data input from EEPROM
EE_SK	B20	T/S	EEPROM serial clock Operates at ~1MHz.
EE_CS_N	B21	T/S	EEPROM chip select Output

2.1.3 System Management Bus (SMB) Interface

The AC specification for these pins is described in Section 11.4.3.3.

Table 2-4. SMB Interface

Symbol	Ball #	Type	Name and Function
SMBD	AD21	T/S, O/D	SMB Data. Stable during the high period of the clock (unless it is a start or stop condition).
SMBCLK	AC21	T/S, O/D	SMB Clock. One clock pulse is generated for each data bit transferred.
SMBALRT_N	AD20	T/S, O/D	SMB Alert: acts as an Interrupt pin of a slave device on the SMB

2.1.4 NC-SI Interface Pins

The AC specification for these pins is described in Section 11.4.3.6.

Table 2-5. NC-SI Interface Pins

Symbol	Ball #	Type	Name and Function
NC-SI_CLK_IN	B5	NC-SI-In	NC-SI Reference Clock Input – Synchronous clock reference for receive, transmit and control interface. It is a 50MHz clock +/-50 ppm.
NC-SI_CLK_OUT	B4	NC-SI-Out	NC-SI Reference Clock Output – Synchronous clock reference for receive, transmit and control interface. It is a 50MHz clock +/-50 ppm. Serves as a clock source to the BMC and the 82576 (when configured so).
NC-SI_CRS_DV	A4	NC-SI-Out	CRS/DV – Carrier Sense / Receive Data Valid.
NC-SI_RXD_1 NC-SI_RXD_0	A6 B7	NC-SI-Out	Receive Data – Data signals from the 82576 to BMC.
NC-SI_TX_EN	B6	NC-SI-In	Transmit Enable.
NC-SI_TXD_1 NC-SI_TXD_0	A7 B8	NC-SI-In	Transmit Data – Data signals from BMC to the 82576.
NC-SI_ARB_OUT	B3	NC-SI-Out/ NC-SI-In	NC-SI HW arbitration token output pin
NC-SI_ARB_IN	AD3	NC-SI-In	NC-SI HW arbitration token input pin

2.1.5 Miscellaneous Pins

The AC specification for the XTAL pins is described in sections 11.4.6.

Table 2-7. Miscellaneous Pins

Symbol	Ball #	Type	Name and Function
SDP0_0 SDP0_1 SDP0_2 SDP0_3	A16 B16 B17 B15	T/S	SW Defined Pins for function 0: These pins are reserved pins that are software programmable w/rt input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. The SDP0[0] pin can be used as a watchdog output indication. All the SDP pins can be used as SFP sideband signals (TxDisable, present & TxFault). The 82576 does not use these signals; it is available for SW control over SFP.
SDP1_0 SDP1_1 SDP1_2 SDP1_3	AD10 A12 A13 AC10	T/S NC-SI T/S T/S	SW Defined Pins for function 1: Reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. The SDP1[0] pin can be used as a watchdog output indication. All the SDP pins can be used as SFP sideband signals (TxDisable, present & TxFault). The 82576 does not use these signals; it is available for SW control over SFP.
MAIN_PWR_OK	AD4	In	Main Power OK – Indicates that platform main power is up. Must be connected externally to main core 3.3V power.
DEV_OFF_N	B9	In	Device Off: Assertion of DEV_OFF_N puts the device in Device Disable mode. This pin is asynchronous and is sampled once the EEPROM is ready to be read following power-up. The DEV_OFF_N pin should always be connected to VCC3P3 to enable device operation.
XTAL1 XTAL2	N23, N24	A-In A-out	Reference Clock / XTAL : These pins may be driven by an external 25MHz crystal or driven by a single ended external CMOS compliant 25MHz oscillator.

2.1.6 SERDES/SGMII Pins

The AC specification for these pins is described in Section 11.4.4.

Table 2-8. SERDES/SGMII Pins

Symbol	Ball #	Type	Name and Function
SRDSI_0_p SRDSI_0_n	J23 J24	A-in	SERDES/SGMII Serial Data input Port 0: Differential SERDES Receive interface. A Serial differential input pair running at 1.25Gb/s. An embedded clock present in this input is recovered along with the data.
SRDSO_0_p SRDSO_0_n	K23 K24	A-out	SERDES/SGMII Serial Data output Port 0: Differential SERDES Transmit interface. A serial differential output pair running at 1.25Gb/s. This output carries both data and an embedded 1.25GHz clock that is recovered along with data at the receiving end.
SRDS_0_SIG_DET	A9	In	Port 0 Signal Detect: Indicates that signal (light) is detected from the Fiber. High for signal detect, Low otherwise.
SRDSI_1_p SRDSI_1_n	T23 T24	A-in	SERDES/SGMII Serial Data input Port 1: Differential fiber SERDES Receive interface. A Serial differential input pair running at 1.25Gb/s. An embedded clock present in this input is recovered along with the data.
SRDSO_1_p SRDSO_1_n	R23 R24	A-out	SERDES/SGMII Serial Data output Port 1: Differential fiber SERDES Transmit interface. A serial differential output pair running at 1.25Gb/s. This output carries both data and an embedded 1.25GHz clock that is recovered along with data at the receiving end.
SRDS_1_SIG_DET	A10	In	Port 1 Signal Detect: Indicates that signal (light) is detected from the fiber. High for signal detect, Low otherwise.
SER_RCOMP	L22	B	Impedance compensation. Connect to ground through an external 1.4 Kohm 1% 100ppm resistor for impedance compensation. See Figure 11-13 for details.

2.1.7 SFP Pins

The AC specification for these pins is described in Chapter 11..

Table 2-9. SFP Pins

Symbol	Ball #	Type	Name and Function
SFP0_I2C_CLK/ MDC0	AD19	Out	Port 0 SFP 2 wire interface clock – connects to Mod-Def1 input of SFP. Can also be used as MDC pin.
SFP0_I2C_DATA/ MDIO0	AD18	T/S, O/D	Port 0 SFP 2 wire interface data – connects to Mod-Def2 pin of SFP. Can also be used as MDIO pin.
SFP1_I2C_CLK/ MDC1	AC19	Out	Port 1 SFP 2 wire interface clock – connects to Mod-Def1 input of SFP. Can also be used as MDC pin.
SFP1_I2C_DATA/ MDIO	AC18	T/S, O/D	Port 1 SFP 2 wire interface data – connects to Mod-Def2 pin of SFP. Can also be used as MDIO pin.

2.1.8 Media Dependent Interface (PHY's MDI) Pins

2.1.8.1 LED's (8)

The table below describes the functionality of the LED output pins. Default activity of the LED may be modified in the EEPROM words 1Ch and 1Fh. The LED functionality is reflected and can be further modified in the configuration registers LEDCTL.

Table 2-11. LED Output Pins

Symbol	Ball #	Type	Name and Function
LED0_0	A19	Out	Port 0 LED0. Programmable LED which indicates by default Link Up.
LED0_1	B19	Out	Port 0 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering).
LED0_2	B18	Out	Port 0 LED2. Programmable LED which indicates by default a 100Mbps Link.
LED0_3	A18	Out	Port 0 LED3. Programmable LED which indicates by default a 1000Mbps Link.

Table 2-11. LED Output Pins (Continued)

LED1_0	AD13	Out	Port 1 LED0. Programmable LED which indicates by default Link up.
LED1_1	AC11	Out	Port 1 LED1. Programmable LED which indicates by default actvity (when packets are transmitted or received that match MAC filtering).
LED1_2	AC13	Out	Port 1 LED2. Programmable LED which indicates by default a 100Mbps Link.
LED1_3	AC12	Out	Port 1 LED3. Programmable LED which indicates by default a 1000Mbps Link.

2.1.8.2 Analog Pins

The AC specification for these pins is described in sections Chapter 11..

Table 2-12. Analog Pins

Symbol	Ball #	Type	Name and Function
MDI0_p_0 MDI0_n_0 MDI1_p_0 MDI1_n_0	C24 C23 AB24 AB23	A	Media Dependent Interface[0] for port 0 and port 1 accordingly: 1000BASE-T: In MDI configuration, MDI[0]+/- corresponds to BI_DA+/- and in MDIX configuration MDI[0]+/- corresponds to BI_DB+/-. 100BASE-TX: In MDI configuration, MDI[0]+/- is used for the transmit pair and in MDIX configuration MDI[0]+/- is used for the receive pair. 10BASE-T: In MDI configuration, MDI[0]+/- is used for the transmit pair and in MDIX configuration MDI[0]+/- is used for the receive pair.
MDI0_p_1 MDI0_n_1 MDI1_p_1 MDI1_n_1	D24 D23 AA24 AA23	A	Media Dependent Interface[1] for port 0 and port 1 accordingly: 1000BASE-T: In MDI configuration, MDI[1]+/- corresponds to BI_DB+/- and in MDIX configuration MDI[1]+/- corresponds to BI_DA+/-. 100BASE-TX: In MDI configuration, MDI[1]+/- is used for the receive pair and in MDIX configuration MDI[1]+/- is used for the transmit pair. 10BASE-T: In MDI configuration, MDI[1]+/- is used for the receive pair and in MDIX configuration MDI[1]+/- is used for the transmit pair.
MDI0_p_2 MDI0_n_2 MDI0_p_3 MDI0_n_3 MDI1_p_2 MDI1_n_2 MDI1_p_3 MDI1_n_3	F24 F23 G24 G23 W24 W23 V24 V23	A	Media Dependent Interface[3:2] for port 0 and port 1: 1000BASE-T: In MDI configuration, MDI[3:2]+/- corresponds to BI_DA+/- and in MDIX configuration MDI[3:2]+/- corresponds to BI_DB+/-. 100BASE-TX: Unused. 10BASE-T: Unused.
IEEE_ATEST0_p IEEE_ATEST0_n IEEE_ATEST1_p IEEE_ATEST1_n	A22 B22 AD22 AC22	A-out	Analog test pins output, used for validation of the transmitter clock jitter.
RBIAS0	E22	B	Compensation reference resistor for LAN0. Connect to ground through an external 1.4 Kohm 1% 100ppm resistor. See Figure 11-12 for details.
RBIAS1	Y22	B	Compensation reference resistor for LAN1. Connect to ground through an external 1.4 Kohm 1% 100ppm resistor. See Figure 11-12 for details.

2.1.9 Testability Pins

Table 2-13. Testability Pins

Symbol	Ball #	Type	Name and Function
JTCK	AC6	In	JTAG Clock Input
JTDI	AD7	In	JTAG TDI Input
JTDO	AC8	O/D	JTAG TDO Output
JTMS	AC7	In	JTAG TMS Input
AUX_PWR	B14	T/S	Auxiliary Power Available: When set, indicates that Auxiliary Power is available and the device should support D3COLD power state if enabled to do so. This pin is also used for testing and scan.
LAN1_DIS_N	A15	T/S	This pin is a strapping option pin latched at the rising edge of PE_RST# or In-Band PCIe* Reset. This pin has an internal weak pull-up resistor. In case this pin is not connected or driven hi during init time, LAN 1 is enabled. In case this pin is driven low during init time, LAN 1 function is disabled. This pin is also used for testing and scan.
LAN0_DIS_N	B13	T/S	This pin is a strapping option pin latched at the rising edge of PE_RST# or In-Band PCIe* Reset. This pin has an internal weak pull-up resistor. In case this pin is not connected or driven hi during init time, LAN 0 is enabled. In case this pin is driven low during init time, LAN 0 function is disabled. This pin is also used for testing and scan.

2.1.10 Reserved Pins and No-Connects

Table 2-14. Reserved Pins and No-Connects

Symbol	Ball #	
RSVDAB18_NC	AB18	Reserved, no-connect. These pins are reserved by Intel and may have factory test functions. For normal operation, do not connect any circuitry to these pins. Do not connect pull-up or pull-down resistors.
RSVDAB19_NC	AB19	
RSVDAC16_NC	AC16	
RSVDAC17_NC	AC17	
RSVDAD16_NC	AD16	
RSVDAD17_NC	AD17	
RSVDM2_NC	M2	
RSVDM23_NC	M23	
RSVDM24_NC	M24	
RSVDM3_NC	M3	
RSVDA8_3P3	A8	Reserved, VCC3P3. These pins are reserved by Intel and may have factory test functions. For normal operation, connect them directly to VCC3P3. Do not connect them to pull-up resistors.
RSVDA11_3P3	A11	
RSVDB10_3P3	B10	
RSVDB11_3P3	B11	
RSVDB12_3P3	B12	
RSVDAD9_3P3	AD9	
RSVDAC5_3P3	AC5	Reserved, VCC3P3. These pins are reserved by Intel and may have factory test functions. For normal operation, connect directly to VCC3P3 with a 10k ohm pull-up resister.
RSVDL14_1P0	L14	Reserved, VCC1P0. These pins are reserved by Intel and may have factory test functions. For normal operation, connect them directly to VCC1P0. Do not connect them to pull-up resistors.
RSVDP14_1P0	P14	

Table 2-14. Reserved Pins and No-Connects (Continued)

RSVDAD8_VSS	AD8	Reserved, VSS. These pins are reserved by Intel and may have factory test functions. For normal operation, connect them directly to VSS. Do not connect them to pull-down resistors.
RSVDA14_VSS	A14	
NCAC3	AC3	Reserved, no connect. This pin is not connected internally.

2.1.11 Power Supply Pins

Table 2-15. Power Supply Pins

Symbol	Ball #	Type	Name and Function
VCC3P3	AD6, AD12	3.3V	3.3V power
VCC3P3	A5, A17	3.3V	3.3V power
VCC1P0	R14, R13, R12, R11, P13, P12, L13, L12, K14, K13, K12, K11	1.0V	1V power digital
VCC1P8	P9, P8,P5, P4, N9, N8, N5, N4, M9, M8, M5, M4, L9, L8, L5, L4	1.8V	1.8V analog power input PCIe*
VCC1P8	L15, K15, J15, H15, G15, E20, E19, D20, D19, AA20, AA19,Y20, Y19, V15, U15, T15, R15, P15, N21, N15, M21, M15	1.8V	1.8V analog power input PHY
VCC1P0	V5, V4, U5, U4, P11, N11, M11, L11, H5, H4, G5, G4	1.0V	1.0V analog power input PCIe*
VCC1P0	J21, J20, J18, J17, L21, L20, L18, L17, K21, K20, K18, K17, T21, T20, T18, T17, P21, P20, P18, P17, R21, R20, R18, R17	1.0V	1.0V analog power input PHY
VSS	Y9, Y8, Y7, Y6, Y15, Y14, Y13, Y12, Y11, Y10, W9, W8, W7, W14, W13, W12, W11, W10, V9, V8, V14, V13, V12, V11, V10, U9, U14, U13, U12, U11, U10, T14, T13, T12, T11, N14, N13, N12, M14, M13, M12, J14, J13, J12, J11, H9, H14, H13, H12, H11, H10, G9, G8, G14, G13, G12, G11, G10, F9, F8, F7, F14, F13, F12, F11, F10, E9, E8, E7, E6, E15, E14, E13, E12, E11, E10, D9, D8, D7, D6, D5, D16, D15, D14, D13, D12, D11, D10, C9, C8, C7, C6, C5, C4, C17, C16, C15, C14, C13, C12, C11, C10, B2, B1, AD5, AD2, AD11, AD1, AC4, AC2, AC1, AB9, AB8, AB7, AB6, AB5, AB4, AB17, AB16, AB15, AB14, AB13, AB12, AB11, AB10, AA9, AA8, AA7, AA6, AA5, AA16, AA15, AA14, AA13, AA12, AA11, AA10, A3, A2, A1	0 V	Digital Ground
VSS	Y24, Y23, Y21, Y18, Y17, Y16, W22, W21, W20, W19, W18, W17, W16, W15, V22, V21, V20, V19, V18, V17, V16, U24, U23, U22, U21, U20, U19, U18, U17, U16, T22, T19, T16, R22, R19, R16, P24, P23, P22, P19, P16, N22, N20, N19, N18, N17, N16, M22, M20, M19, M18, M17, M16, L24, L23, L19, L16, K22, K19, K16, J22, J19, J16, H24, H23, H22, H21, H20, H19, H18, H17, H16, G22, G21, G20, G19, G18, G17, G16, F22, F21, F20, F19, F18, F17, F16, F15, E24, E23, E21, E18, E17, E16, D22, D21, D18, D17, C22, C21, C20, C19, C18, B24, B23, AD24, AD23, AC24, AC23, AB22, AB21, AB20, AA22, AA21, AA18, AA17, A24, A23	0 V	PHY analog ground

Table 2-15. Power Supply Pins (Continued)

VSS	Y5, Y4, Y3, Y2, Y1, W6, W5, W4, W3, V7, V6, V3, V2, V1, U8, U7, U6, U3, T9, T8, T7, T6, T5, T4, T3, T2, T10, T1, R9, R8, R7, R6, R5, R4, R3, R10, P7, P6, P3, P2, P10, P1, N7, N6, N3, N10, M7,M6, M10, M1, L7, L6, L3, L2, L10, K9, K8, K7, K6, K5, K4, K3, K10, J9, J8, J7, J6, J5, J4, J3, J2, J10, J1, H8, H7, H6, H3, G7, G6, G3, G2, G1, F6, F5, F4, F3, E5, E4, E3, E2, E1, D4, D3, C3, C2, C1, AB3, AB2, AB1, AA4, AA3	0V	PCIe* analog ground

2.2 Pull-ups/Pull-downs

The table below lists internal & external pull-up resistors and their functionality in different device states.

Each internal PUP has a nominal value of 5KΩ, ranging from 2.7KΩ to 8.6KΩ.. The recommended values for external resistors are 400Ω for pull down resistors and 3KΩ for pull up resistors.

The device states are defined as follow:

- Power-up = while 3.3V is stable, yet 1.0V isn't
- Active = normal mode (not power up or disable)
- Disable = device disable (a.k.a. dynamic IDDQ – see See Section 4.4)

Table 2-16. Pull-Up Resistors

Signal Name	Power up		Active		Disable		External
	PUP	Comments	PUP	Comments	PUP	Comments	
PE_WAKE_N	N		N		N		Y
PE_RST_N	Y		N		N		N
FLSH_SI	Y		N		Y		N
FLSH_SO	Y		Y		Y		N
FLSH_SCK	Y		N		Y		N
FLSH_CE_N	Y		N		Y		N
EE_DI	Y		N		Y		N
EE_DO	Y		Y		Y		N
EE_SK	Y		N		Y		N
EE_CS_N	Y		N		Y		N
SMBD	N		N		N		Y
SMBCLK	N		N		N		Y
SMBALRT_N	N		N		N		Y
RSVDAD17_NC	Y		N		N		N
RSVDAC17_NC	Y		N		N		N
RSVDAC16_NC	Y		N		Y	HiZ	N
RSVDAD16_NC	Y		N		Y	HiZ	N
NC-SI_CLK_IN	N	HiZ	N		N		PD (Note 1)
NC-SI_CLK_OUT	Y	HiZ	N		N	If active, stable output	N

Table 2-16. Pull-Up Resistors (Continued)

Signal Name	Power up		Active		Disable		External
	PUP	Comments	PUP	Comments	PUP	Comments	
NC-SI_CRS_DV	N	HiZ	N		N		PD
NC-SI_RXD[1:0]	Y	HiZ	N		N		Y (Note 2)
NC-SI_TX_EN	N	HiZ	N		N		PD (Note 1)
NC-SI_TXD[1:0]	N	HiZ	N		N		PD (Note 1)
NC-SI_ARB_IN	N		Y	Controlled by EEPROM	Y	Controlled by EEPROM	
NC-SI_ARB_OUT	Y		Y		Y		
SDP0[3:0]	Y		Y	Until EEPROM done	N	May keep state by EEPROM control	N
SDP1[3:0]	Y		Y	Until EEPROM done	N		N
DEV_OFF_N	Y		N		N		Must be connected on board
MAIN_PWR_OK	Y		N		N		Must be connected on board
SRDS_0_SIG_DET	Y		N		N		Must be connected externally
SRDS_1_SIG_DET	Y		N		N		Must be connected externally
SFP0_I2C_CLK	Y		N		Y		Y if active
SFP0_I2C_DATA	Y		N		N		Y
SFP1_I2C_CLK	Y		N		Y		Y if active
SFP1_I2C_DATA	Y		N		N		Y
LED0_0	Y		N		N	HiZ	
LED0_1	Y		N		N	HiZ	
LED0_2	Y		N		N	HiZ	
LED0_3	Y		N		N	HiZ	
LED1_0	Y		N		N	HiZ	
LED1_1	Y		N		N	HiZ	
LED1_2	Y		N		N	HiZ	
LED1_3	Y		N		N	HiZ	
JTCK	Y		N		N		N
JTDI	Y		N		N		Y
JTDO	Y		N		N		Y
JTMS	Y		N		N		Y
AUX_PWR	Y		N		N		PU or PD (Note 3)

Table 2-16. Pull-Up Resistors (Continued)

Signal Name	Power up		Active		Disable		External
	PUP	Comments	PUP	Comments	PUP	Comments	
LAN1_DIS_N	Y		Y when input		Y		PU or PD (Note 4)
LAN0_DIS_N	Y		Y when input		Y		PU or PD (Note 4)

Notes:

1. Should be pulled down if NC-SI interface is disabled.
2. Only if NC-SI is unused or set to multi drop configuration.
3. If Aux power is connected, should be pulled up, else should be pulled down.
4. If the specific function is disabled, should be pulled down, else should be pulled up.

2.3 Strapping

The following signals are used for static configuration. Unless otherwise stated, strapping options are latched on the rising edge of Internal_Power_On_Reset, at power up, at in-band PCI Express reset and at PE_RST_N assertion. At other times, they revert to their standard usage.

Table 2-17. Strapping Options

Purpose	Pin	Polarity	Pull-up / Pull-down
LAN1 Disable	LAN1_Dis_N	0b – LAN1 is disabled 1b – LAN1 is enabled	Internal pull-up
LAN0 Disable	LAN1_Dis_N	0b – LAN0 is disabled 1b – LAN0 is enabled	Internal pull-up
AUX_PWR	AUX_PWR	0b – AUX power is not available 1b – AUX power is available	None

2.4 Interface Diagram

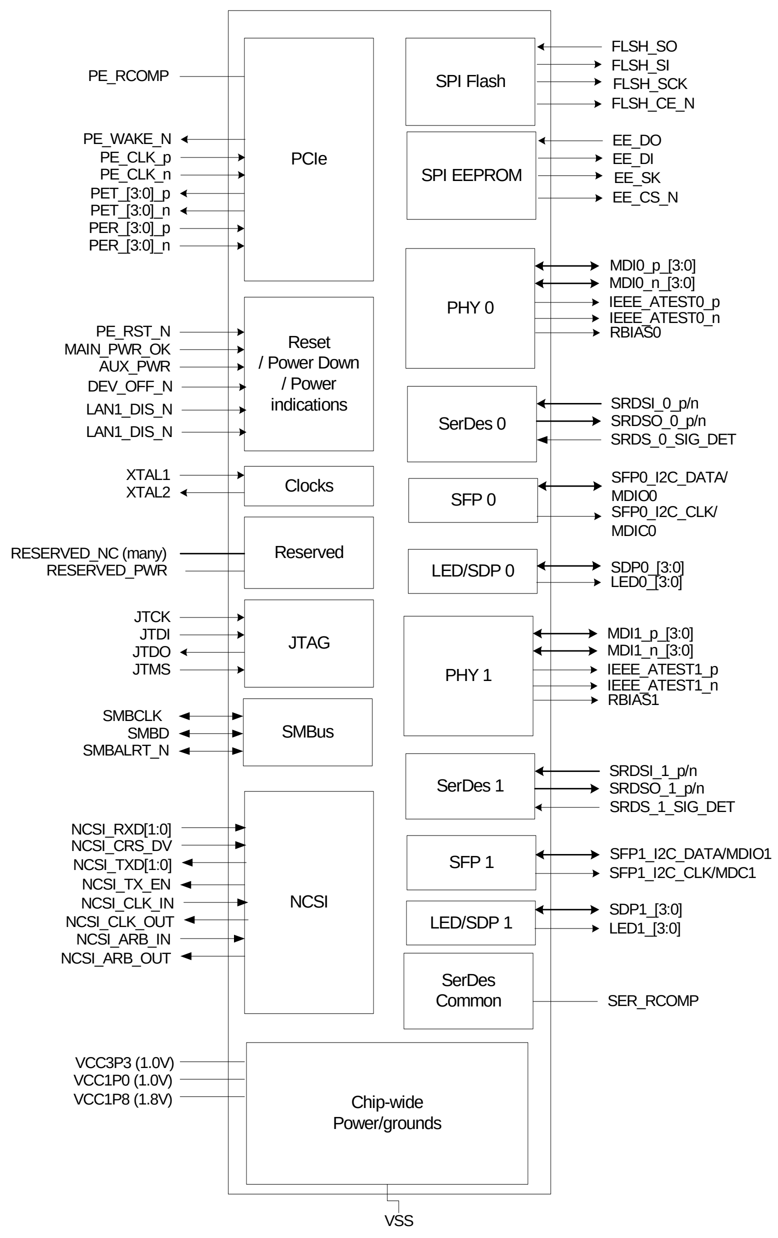


Figure 2-1. Intel® 82576 GbE Controller Interface Diagram

2.5 Pin List (Alphabetical)

Table 2-18 lists the pins and signals in pin alphabetical order. Note that where multiple pins are listed, the list sorts by the lowest pin designator. VSS pins are in Table 2-19.

Table 2-18. Pin List (Alphabetical by Pin Designation)

Signal	Pin	Signal	Pin	Signal	Pin
NC-SI_CRS_DV	A4	LED0_2	B18	RSVDM2_NC	M2
VCC3P3	A5, A17	LED0_1	B19	RSVDM3_NC	M3
NC-SI_RXD_1	A6	EE_SK	B20	RSVDM23_NC	M23
NC-SI_TXD_1	A7	EE_CS_N	B21	RSVDM24_NC	M24
RSVDA8_3P3	A8	IEEE_ATEST0_n	B22		
SRDS_0_SIG_DET	A9			PE_CLK_n	N1
SRDS_1_SIG_DET	A10	MDI0_n_0	C23	PE_CLK_p	N2
RSVDA11_3P3	A11	MDI0_p_0	C24	XTAL1	N23
SDP1_1	A12	PET_0_n	D1	XTAL2	N24
SDP1_2	A13	PET_0_p	D2	VCC1P8	P9, P8,P5, P4, N9, N8, N5, N4, M9, M8, M5, M4, L9, L8, L5, L4
RSVDA14_VSS	A14	MDI0_n_1	D23	RSVDP14_1P0	P14
LAN1_DIS_N	A15	MDI0_p_1	D24		
SDP0_0	A16	RBIAS0	E22	PET_2_n	R1
LED0_3	A18	PER_0_n	F1	PET_2_p	R2
LED0_0	A19	PER_0_p	F2	VCC	R14, R13, R12, R11, P13, P12, L13, L12, K14, K13, K12, K11
EE_DO	A20	MDI0_n_2	F23	SRDSO_1_p	R23
EE_DI	A21	MDI0_p_2	F24	SRDSO_1_n	R24
IEEE_ATEST0_p	A22	MDI0_n_3	G23	SRDSI_1_p	T23
NC-SI_ARB_OUT	B3	MDI0_p_3	G24	SRDSI_1_n	T24
NC-SI_CLK_OUT	B4	PET_1_n	H1	PER_2_n	U1
NC-SI_CLK_IN	B5	PET_1_p	H2	PER_2_p	U2
NC-SI_TX_EN	B6	VCC1P0	J21, J20, J18, J17, L21, L20, L18, L17, K21, K20, K18, K17, T21, T20, T18, T17, P21, P20, P18, P17, R21, R20, R18, R17	MDI1_n_3	V23
NC-SI_RXD_0	B7	SRDSI_0_p	J23	MDI1_p_3	V24
NC-SI_TXD_0	B8	SRDSI_0_n	J24	VCC1P0	V5, V4, U5, U4, P11, N11, M11, L11, H5, H4, G5, G4
DEV_OFF_N	B9	PER_1_n	K1		
RSVDB10_3P3	B10	PER_1_p	K2		
RSVDB11_3P3	B11	SRDSO_0_p	K23	PET_3_n	W1
RSVDB12_3P3	B12	SRDSO_0_n	K24	PET_3_p	W2
LAN0_DIS_N	B13	PE_RCOMP	L1	MDI1_n_2	W23

Table 2-18. Pin List (Alphabetical by Pin Designation) (Continued)

Signal	Pin	Signal	Pin	Signal	Pin
AUX_PWR	B14	VCC1P8	P9, P8,P5, P4, N9, N8, N5, N4, M9, M8, M5, M4, L9, L8, L5, L4	MDI1_p_2	W24
SDP0_3	B15	RSVDL14_1P0	L14	RBIAS1	Y22
SDP0_1	B16	VCC1P8	L15, K15, J15, H15, G15, E20, E19, D20, D19, AA20, AA19,Y20, Y19, V15,U15, T15, R15, P15, N21, N15, M21, M15		
SDP0_2	B17	SER_RCOMP	L22		
PER_3_n	AA1	SDP1_3	AC10	VCC3P3	AD6, AD12
PER_3_p	AA2	LED1_1	AC11	JTDI	AD7
MDI1_n_1	AA23	LED1_3	AC12	RSVDAD8_VSS	AD8
MDI1_p_1	AA24	LED1_2	AC13	RSVDAD9_3P3	AD9
RSVDAB18_NC	AB18	FLSH_SI	AC14	SDP1_0	AD10
RSVDAB19_NC	AB19	FLSH_CE_N	AC15	LED1_0	AD13
MDI1_n_0	AB23	SFP1_I2C_DATA/ MDIO1	AC18	FLSH_SO	AD14
MDI1_p_0	AB24	SFP1_I2C_CLK/ MDC1	AC19	FLSH_SCK	AD15
NCAC3	AC3	PE_WAKE_N	AC20	SFP0_I2C_DATA /MDIO0	AD18
RSVDAC5_3P3	AC5	SMBCLK	AC21	SFP0_I2C_CLK/ MDC0	AD19
JTCK	AC6	IEEE_ATEST1_n	AC22	SMBALRT_N	AD20
JTMS	AC7			SMBD	AD21
JTDO	AC8	NC-SI_ARB_IN	AD3	IEEE_ATEST1_p	AD22
PE_RST_N	AC9	MAIN_PWR_OK	AD4		

Table 2-19. VSS Pins

Signal	Pin
VSS	Y24, Y23, Y21, Y18, Y17, Y16, W22, W21, W20, W19, W18, W17, W16, W15, V22, V21, V20, V19, V18, V17, V16, U24, U23, U22, U21, U20, U19, U18, U17, U16, T22, T19, T16, R22, R19, R16, P24, P23, P22, P19, P16, N22, N20, N19, N18, N17, N16, M22, M20, M19, M18, M17, M16, L24, L23, L19, L16, K22, K19, K16, J22, J19, J16, H24, H23, H22, H21, H20, H19, H18, H17, H16, G22, G21, G20, G19, G18, G17, G16, F22, F21, F20, F19, F18, F17, F16, F15, E24, E23, E21, E18, E17, E16, D22, D21, D18, D17, C22, C21, C20, C19, C18, B24, B23, AD24, AD23, AC24, AC23, AB22, AB21, AB20, AA22, AA21, AA18, AA17, A24, A23, Y5, Y4, Y3, Y2, Y1, W6, W5, W4, W3, V7, V6, V3, V2, V1, U8, U7, U6, U3, T9, T8, T7, T6, T5, T4, T3, T2, T10, T1, R9, R8, R7, R6, R5, R4, R3, R10, P7, P6, P3, P2, P10, P1, N7, N6, N3, N10, M7,M6, M10, M1, L7, L6, L3, L2, L10, K9, K8, K7, K6, K5, K4, K3, K10, J9, J8, J7, J6, J5, J4, J3, J2, J10, J1, H8, H7, H6, H3, G7, G6, G3, G2, G1, F6, F5, F4, F3, E5, E4, E3, E2, E1, D4, D3, C3, C2, C1, AB3, AB2, AB1, AA4, AA3

2.6 Ball Out

This section provides a top view ball map of the 82576 in a 25 mmx25 mm package. Some names in the layout are not accurate (short names were chosen to fit). See Figure 2-2 for the color key for the ball out table.

🟦	Clock/BIAS/IEEE test pins	🟥	MDI Interface NC-SI Signals
🟩	VCC1P8 VCC1P0	▒	VSS
⬜	Functional Pin	🟦	PCIe signals
🟨	VCC3P3	🟧	Open Drain
🟪	Reserved signals	🟩	MDIO/2 Wire Interface signals

Figure 2-2. Color Key for Ball-Out

	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
AD	VSS	VSS	IEEE_ATEST1_p	SMBD	SMBALRT_N	SFP0_I2C_CLK/MDC0	SFP0_I2C_Data/MDIO0	RSVDAD17_NC	RSVDAD16_NC	FLSH_SCK	FLSH_SO	LED1_0	VCC3P3	VSS	SDP1_0	RSVDAD9_3P3	RSVDAD8_VSS	JTDI	VCC3P3	VSS	MAIN_PWR_OK	NCSI_ARB_IN	VSS	VSS
AC	VSS	VSS	IEEE_ATEST1_n	SMBCLK	PE_WAKE_N	SFP1_I2C_CLK/MDC1	SFP1_I2C_Data/MDIO1	RSVDAC17_NC	RSVDAC16_NC	FLSH_CE_N	FLSH_SI	LED1_2	LED1_3	LED1_1	SDP1_3	PE_RST_N	JTDO	JTMS	JTCK	RSVDAC5_3P3	VSS	NCAC3	VSS	VSS
AB	MDI1_p_0	MDI1_n_0	VSS	VSS	VSS	RSVDAB19_NC	RSVDAB18_NC	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
AA	MDI1_p_1	MDI1_n_1	VSS	VSS	VCC1P8	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	PER_3_p	PER_3_n
Y	VSS	VSS	RBIAS1	VSS	VCC1P8	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
W	MDI1_p_2	MDI1_n_2	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	PET_3_p	PET_3_n
V	MDI1_p_3	MDI1_n_3	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1p0_PE	VCC1p0_PE	VSS	VSS	VSS
U	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1p0_PE	VCC1p0_PE	VSS	PER_2_p	PER_2_n
T	SRDSI_1_n	SRDSI_1_p	VSS	VCC1P0	VCC1P0	VSS	VCC1P0	VCC1P0	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
R	SRDSO_1_n	SRDSO_1_p	VSS	VCC1P0	VCC1P0	VSS	VCC1P0	VCC1P0	VSS	VCC1P8	VCC1P0	VCC1P0	VCC1P0	VCC1P0	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	PET_2_p	PET_2_n
P	VSS	VSS	VSS	VCC1P0	VCC1P0	VSS	VCC1P0	VCC1P0	VSS	VCC1P8	RSVDP14_NC	VCC1P0	VCC1P0	VCC1P0	VSS	VCC1P8	VCC1P8	VSS	VSS	VCC1P8	VCC1P8	VSS	VSS	VSS
N	XTAL2	XTAL1	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VCC1P8	VSS	VSS	VSS	VCC1P0	VSS	VCC1P8	VCC1P8	VSS	VSS	VCC1P8	VCC1P8	VSS	PE_CLK_p	PE_CLK_n
M	RSVDM24_NC	RSVDM23_NC	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VCC1P8	VSS	VSS	VSS	VCC1P0	VSS	VCC1P8	VCC1P8	VSS	VSS	VCC1P8	VCC1P8	RSVDM3_NC	RSVDM2_NC	VSS
L	VSS	VSS	SER_RCOMP	VCC1P0	VCC1P0	VSS	VCC1P0	VCC1P0	VSS	VCC1P8	RSVDL14_1P0	VCC1P0	VCC1P0	VCC1P0	VSS	VCC1P8	VCC1P8	VSS	VSS	VCC1P8	VCC1P8	VSS	VSS	PE_RCOMP
K	SRDSO_0_n	SRDSO_0_p	VSS	VCC1P0	VCC1P0	VSS	VCC1P0	VCC1P0	VSS	VCC1P8	VCC1P0	VCC1P0	VCC1P0	VCC1P0	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	PER_1_p	PER_1_n
J	SRDSI_0_n	SRDSI_0_p	VSS	VCC1P0	VCC1P0	VSS	VCC1P0	VCC1P0	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
H	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P0	VCC1P0	VSS	PET_1_p	PET_1_n
G	MDI0_p_3	MDI0_n_3	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P0	VCC1P0	VSS	VSS	VSS
F	MDI0_p_2	MDI0_n_2	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	PER_0_p	PER_0_n
E	VSS	VSS	RBIAS0	VSS	VCC1P8	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
D	MDI0_p_1	MDI0_n_1	VSS	VSS	VCC1P8	VCC1P8	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	PET_0_p	PET_0_n
C	MDI0_p_0	MDI0_n_0	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
B	VSS	VSS	IEEE_ATEST0_n	EE_CS_N	EE_SK	LED0_1	LED0_2	SDP0[2]	SDP0[1]	SDP0[3]	AUX_PWR	LAN0_Dis_N	RSVDB12_NC	RSVDB11_NC	RSVDB10_NC	DEV_OFF_N	NCSI_TXD[0]	NCSI_RXD[	NCSI_TX_E	NCSI_CLK_	NCSI_ CLK	NCSI_ARB_OUT	VSS	VSS
A	VSS	VSS	IEEE_ATEST0_p	EE_DI	EE_DO	LED0_0	LED0_3	VCC3P3	SDP0[0]	LAN1_DIS_N	RSVDA14_NC	SDP1[2]	SDP1[1]	RSVDA11_NC	SRDS1_SIG_ DET	SRDS0_SIG_DET	RSVDA8_NC	NCSI_TXD[	NCSI_RXD[	VCC3P3	NCSI_CRS_	VSS	VSS	VSS

Figure 2-3. Ball-Out Representation

§ §

NOTE: ***This page intentionally left blank.***

3. Interconnects

3.1 PCIe

3.1.1 PCIe Overview

PCIe is a third generation I/O architecture that enables cost competitive next generation I/O solutions providing industry leading price/performance and feature richness. It is an industry-driven specification.

PCIe defines a basic set of requirements that encases the majority of the targeted application classes. Higher-end applications' requirements, such as enterprise class servers and high-end communication platforms, are encased by a set of advanced extensions that compliment the baseline requirements.

To guarantee headroom for future applications of PCIe, a software-managed mechanism for introducing new, enhanced, capabilities in the platform is provided. Figure 3-1 shows PCIe architecture.

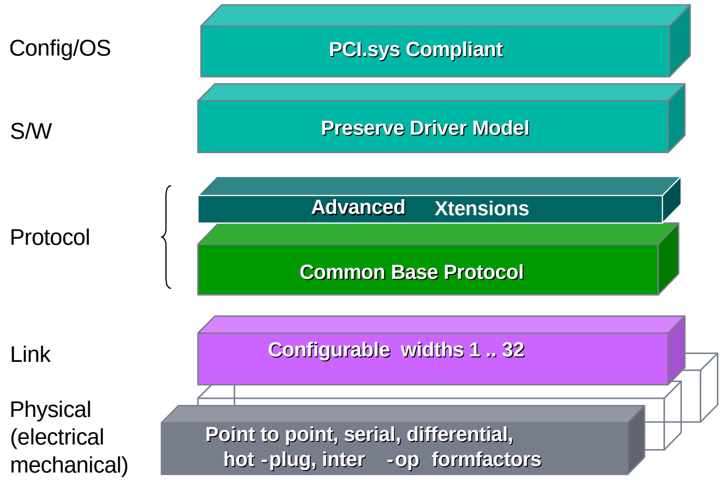

Config/OS
PCI.sys Compliant
S/W
Preserve Driver Model
Protocol
Advanced Xtensions
Common Base Protocol
Link
Configurable widths 1 .. 32
Physical (electrical mechanical)
Point to point, serial, differential, hot -plug, inter -op formfactors

Figure 3-1. PCIe Stack Structure

PCIe's physical layer consists of a differential transmit pair and a differential receive pair. Full-duplex data on these two point-to-point connections is self-clocked such that no dedicated clock signals are required. The bandwidth of this interface increases linearly with frequency.

The packet is the fundamental unit of information exchange and the protocol includes a message space to replace the various side-band signals found on many buses today. This movement of hard-wired signals from the physical layer to messages within the transaction layer enables easy and linear physical layer width expansion for increased bandwidth.

The common base protocol uses split transactions and several mechanisms are included to eliminate wait states and to optimize the reordering of transactions to further improve system performance.

3.1.1.1 Architecture, Transaction and Link Layer Properties

- Split transaction, packet-based protocol
- Common flat address space for load/store access (such as PCI addressing model)
  - — Memory address space of 32-bit to allow compact packet header (must be used to access addresses below 4 GB)
  - — Memory address space of 64-bit using extended packet header
- Transaction layer mechanisms:
  - — PCI-X style relaxed ordering
  - — Optimizations for no-snoop transactions
- Credit-based flow control
- Packet sizes/formats:
  - — Maximum packet size supports 128 byte and 256 byte data payload
  - — Maximum read request size of 512 bytes
- Reset/initialization:
  - — Frequency/width/profile negotiation performed by hardware
- Data integrity support
  - — Using CRC-32 for transaction layer packets
- Link layer retry for recovery following error detection
  - — Using CRC-16 for link layer messages
- No retry following error detection
  - — 8b/10b encoding with running disparity
- Software configuration mechanism:
  - — Uses PCI configuration and bus enumeration model
  - — PCIe-specific configuration registers mapped via PCI extended capability mechanism
- Baseline messaging:
  - — In-band messaging of formerly side-band legacy signals (such as interrupts, etc.)
  - — System-level power management supported via messages
- Power management:
  - — Full support for PCI-PM
  - — Wake capability from D3cold state
  - — Compliant with ACPI, PCI-PM software model
  - — Active state power management
- Support for PCIe v2.0 (2.5GT/s)
  - — Support for completion time out
  - — Support for additional registers in the PCIe capability structure.

3.1.1.2 Physical Interface Properties

- Point to point interconnect
  - — Full-duplex; no arbitration

- Signaling technology:
  - — Low Voltage Differential (LVD)
  - — Embedded clock signaling using 8b/10b encoding scheme
- Serial frequency of operation: 2.5 GHz.
- Interface width of x4, x2, or x1.
- DFT and DFM support for high volume manufacturing

3.1.1.3 Advanced Extensions

PCIe defines a set of optional features to enhance platform capabilities for specific usage modes. The 82576 supports the following optional features:

- Advanced Error Reporting - messaging support to communicate multiple types/severity of errors
- Device serial number - Allows exposure of a unique serial number for each device.
- Alternative Requester ID (ARI) - allow support of more than 8 function per device.
- Single Root I/O virtualization (PCI-SIG SR-IOV) - allows exposure of virtual functions controlling a subset of the resources to virtual machines.

3.1.2 Functionality - General

3.1.2.1 Native/Legacy

- All the 82576 PCI functions are native PCIe functions.

3.1.2.2 Locked Transactions

- The 82576 does not support locked requests as target or master.

3.1.2.3 End to End CRC (ECRC)

- Not supported by the 82576

3.1.3 Host I/F

3.1.3.1 Tag IDs

PCIe device numbers identify logical devices within the physical device (the 82576 is a physical device). The 82576 implements a single logical device with up to two separate PCI functions: LAN 0, and LAN 1. The device number is captured from each type 0 configuration write transaction.

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 enable routing of the completion to the appropriate client.

Tag IDs are allocated differently for read and write. Messages are sent with a tag of 0x1F.

3.1.3.1.1 TAG ID Allocation for Read Transactions

Table 3-1 lists the Tag ID allocation for read accesses. The tag ID is interpreted by hardware in order to forward the read data to the required device.

Table 3-1. IDs in Read Transactions

Tag ID	Description	Comment
0	Reserved	
1	Descriptor Rx	Like 82571/82572/82575
2	Reserved	
3	Reserved	
4	Descriptor Tx	Like 82571/82572/82575
5	Reserved	
6	Reserved	
7	Reserved	
8	Data request 0	Like 82571/82572/82575
9	Data request 1	Like 82571/82572/82575
0a	Data request 2	Like 82571/82572/82575
0b	Data request 3	Like 82571/82572/82575
10	Reserved	
11	Message unit	
12-1F	Reserved	

3.1.3.1.2 TAG ID Allocation for Write Transactions

Request tag allocation depends on these system parameters:

- DCA supported/not supported in the system (*DCA_CTRL.DCA_DIS* - see Section 8.13.4 for details)
- DCA enabled/disabled for each type of traffic (*TXCTL.TX Descriptor DCA EN, RXCTL.RX Descriptor DCA EN, RXCTL.RX Header DCA EN, RXCTL.Rx Payload DCA EN*)
- System type: Legacy DCA vs. DCA 1.0 (*DCA_CTRL.DCA_MODE* - see Section 8.13.4 for details).
- CPU ID (*RXCTL.CPUID* or *TXCTL.CPUID*)

Since DCA is implemented differently in I/OAT 1 and in I/OAT 2/3 platforms, the tag IDs are different as well (see Section 3.1.3.1.2.3 below).

3.1.3.1.2.1 Case 1 - DCA Disabled in the System:

Table 3-2 describes the write requests tags. Unlike read, the values are for debug only, allowing tracing of requests through the system.

Table 3-2. IDs in Write Transactions, DCA Disabled Mode

Tag ID	Description
0x0 - 0x1	Reserved
0x2	Tx descriptors write-back / Tx Head write-back
0x3	Reserved
0x4	Rx descriptors write-back
0x5	Reserved
0x6	Write data

Table 3-2. IDs in Write Transactions, DCA Disabled Mode (Continued)

0x7 - 0x1D	Reserved
0x1E	MSI and MSI-X
0x1F	Reserved

3.1.3.1.2.2 Case 2 - DCA Enabled in the System, but Disabled for the Request:

- Legacy DCA platforms - If DCA is disabled for the request, the tags allocation is identical to the case where DCA is disabled in the system. See Table 3-2 above.
- DCA 1.0 platforms - All write requests have the tag of 0x00.

Note: When in DCA 1.0 mode, messages and MSI/MSI-x write requests are sent with the no-hint tag.

3.1.3.1.2.3 Case 3 - DCA Enabled in the System, DCA Enabled for the Request:

- Legacy DCA Platforms: the request tag is constructed as follows:
  - — Bit[0] – DCA Enable
  - — Bits[3:1] - The CPU ID field taken from the CPUID[2:0] bits of the RXCTL or TXCTL registers
  - — Bits[7:4] - Reserved
- DCA 1.0 Platforms: the request tag (all 8 bits) is taken from the CPUID field of the RXCTL or TXCTL registers

3.1.3.2 Completion Timeout Mechanism

In any split transaction protocol, there is a risk associated with the failure of a requester to receive an expected completion. To enable requesters to attempt recovery from this situation in a standard manner, the completion timeout mechanism is defined.

The completion timeout mechanism is activated for each request that requires one or more completions when the request is transmitted. The PCIe specification, Rev. 1.1 requires that:

- The completion timeout timer should not expire in less than 10 ms.
- The completion timeout timer must expire if a request is not completed within 50 ms.

However, some platforms experience completion latencies that are longer than 50 ms, in some cases up to seconds. The 82576 supports both the legacy Rev. 1.1 and the default Rev 2.0 mechanisms, To support the legacy mode, it provides a programmable range for the completion timeout, as well as the ability to disable completion timeout altogether. The new capability structure is assigned a PCIe capability structure version of 0x2.

The 82576 controls the following aspects of completion timeout:

- Disabling or enabling completion timeout
- Disabling or enabling resending a request on completion timeout
- A programmable range of timeout values

Programming the behavior of the completion timeout is done differently whether capability structure version 0x1 is enabled or capability structure version 0x2 is enabled. Table 3-3 lists the behavior for both cases.

Table 3-3. Completion Timeout Programming

Capability	Capability Structure Version = 0x1	Capability Structure Version = 0x2
Completion timeout enabling	Loaded from EEPROM into CSR bit.	Controlled through PCI configuration. Visible through read-only CSR bit.
Resend request enable	Loaded from EEPROM into CSR bit.	Loaded from EEPROM into read-only CSR bit.
Completion timeout period	Loaded from EEPROM into CSR bit.	Controlled through PCI configuration. Visible through read-only CSR bit.

The capability structure exposed and the mode used are fixed by the GIO_CAP field in the PCIe Init Configuration 3 EEPROM Word (Word 0x1A).

3.1.3.2.1 Completion Timeout Enable

- Version = 0x1- Loaded from the *Completion Timeout Disable* bit in the EEPROM into the *Completion_Timeout_Disable* bit in the PCIe Control Register (GCR). Completion Timeout enabled is the default.
- Version = 0x2 - Programmed through PCI configuration. Visible through the *Completion_Timeout_Disable* bit in the PCIe Control Register (GCR). Completion Timeout enabled is the default.

3.1.3.2.2 Resend Request Enable

- Version = 0x1- The *Completion Timeout Resend* EEPROM bit, loaded to the *Completion_Timeout_Resend* bit in the PCIe Control Register (GCR), enables resending the request (applies only when completion timeout is enabled). The default is to resend a request that timed out.
- Version = 0x2 - same as when version = 0x1.

3.1.3.2.3 Completion Timeout Period

- Version = 0x1.- Loaded from the *Completion Timeout Value* field in the EEPROM to the *Completion_Timeout_Value* bits in the PCIe Control Register (GCR). The following values are supported:
  - — 50 μs to 10 ms (default)
  - — 10 ms to 250 ms
  - — 250 ms to 4 s
  - — 4 s to 64 s
- Version = 0x2 - Programmed through PCI configuration. Visible through the *Completion_Timeout_Value* bits in the PCIe Control Register (GCR). The 82576 supports all four ranges defined by the PCIe ECR:
  - — 50 μs to 10 ms
  - — 10 ms to 250 ms
  - — 250 ms to 4 s
  - — 4 s to 64 s

System software programs a range (one of nine possible ranges that sub-divide the four ranges previously mentioned) into the PCI configuration register. The supported sub-ranges are:

- — 50 μs to 50 ms (default)
- — 50 μs to 100 μs
- — 1 ms to 10 ms

— 16 ms to 55 ms

— 65 ms to 210 ms

— 260 ms to 900 ms

— 1 s to 3.5 s

— 4 s to 13 s

— 17 s to 64 s

A memory read request for which there are multiple completions is considered completed only when all completions are received by the requester. If some, but not all, requested data is returned before the completion timeout timer expires, the requestor is permitted to keep or to discard the data that was returned prior to timer expiration.

3.1.4 Transaction Layer

The upper layer of the PCIe architecture is the transaction Layer. The transaction layer connects to the 82576 core using an implementation specific protocol. Through this core-to-transaction-layer protocol, the application-specific parts of the 82576 interact with the PCIe subsystem and transmit and receive requests to or from the remote PCIe agent, respectively.

3.1.4.1 Transaction Types Accepted by the 82576

Table 3-4. Transaction Types at the Rx Transaction Layer

Transaction Type	FC Type	Tx Later Reaction	Hardware Should Keep Data From Original Packet	For Client
Configuration Read Request	NPH	CPLH + CPLD	Requester ID, TAG, Attribute	Configuration space
Configuration Write Request	NPH + NPD	CPLH	Requester ID, TAG, Attribute	Configuration space
Memory Read Request	NPH	CPLH + CPLD	Requester ID, TAG, Attribute	CSR
Memory Write Request	PH + PD	-	-	CSR
IO Read Request	NPH	CPLH + CPLD	Requester ID, TAG, Attribute	CSR
IO Write Request	NPH + NPD	CPLH	Requester ID, TAG, Attribute	CSR
Read completions	CPLH + CPLD	-	-	DMA
Message	PH	-	-	Message Unit / INT / PM / Error Unit

Flow control types:

- PH - Posted request headers
- PD - Posted request data payload
- NPH - Non-posted request headers
- NPD - Non-posted request data payload
- CPLH - Completion headers
- CPLD - Completion data payload

3.1.4.1.1 Configuration Request Retry Status

PCIe supports devices requiring a lengthy self-initialization sequence to complete before they are able to service configuration requests as it is the case for the 82576 that might have a delay in initialization due to an EEPROM read.

If the read of the PCIe section in the EEPROM was not completed and the 82576 receives a configuration request, the 82576 responds with a configuration request retry completion status to terminate the request, and thus effectively stall the configuration request until such time that the subsystem has completed local initialization and is ready to communicate with the host.

3.1.4.1.2 Partial Memory Read and Write Requests

The 82576 has limited support of read and write requests when only part of the byte enable bits are set as described later in this section.

Partial writes to the MSI-X table are supported. All other partial writes are ignored and a completion abort is sent.

Zero-length writes have no internal impact (nothing written, no effect such as clear-by-write). The transaction is treated as a successful operation (no error event).

Partial reads with at least one byte enabled are answered as a full read. Any side effect of the full read (such as clear by read) is applicable to partial reads also.

Zero-length reads generate a completion, but the register is not accessed and undefined data is returned.

3.1.4.2 Transaction Types Initiated by the 82576

Table 3-5. Transaction Types at the Tx Transaction Layer

Transaction type	Payload Size	FC Type	From Client
Configuration Read Request Completion	Dword	CPLH + CPLD	Configuration space
Configuration Write Request Completion	-	CPLH	Configuration space
I/O Read Request Completion	Dword	CPLH + CPLD	CSR
I/O Write Request Completion	-	CPLH	CSR
Read Request Completion	Dword/Qword	CPLH + CPLD	CSR
Memory Read Request	-	NPH	DMA
Memory Write Request	<= MAX_PAYLOAD_SIZE	PH + PD	DMA
Message	-	PH	Message Unit / INT / PM / Error Unit

Note: MAX_PAYLOAD_SIZE supported is loaded from EEPROM (128 bytes, 256 bytes or 512 bytes). IF ARI capability is not exposed, the effective MAX_PAYLOAD_SIZE is defined for each PCI functions according to configuration space register of this function. If ARI capability is exposed, effective MAX_PAYLOAD_SIZE is defined for all PCI functions according to configuration space register of function zero

3.1.4.2.1 Data Alignment

Requests must never specify an address/length combination that causes a memory space access to cross a 4 KB boundary. The 82576 breaks requests into 4 KB-aligned requests (if needed). This does not pose any requirement on software. However, if software allocates a buffer across a 4 KB boundary, hardware issues multiple requests for the buffer. Software should consider limiting buffer sizes and base addresses to comply with a 4 KB boundary in cases where it improves performance.

The general rules for packet alignment are as follows:

1. The length of a single request should not exceed the PCIe limit of MAX_PAYLOAD_SIZE for write and MAX_READ_REQ for read.
2. The length of a single request does not exceed the 82576's internal limitations 512 bytes.
3. A single request should not span across different memory pages as noted by the 4 KB boundary previously mentioned.

Note: The rules apply to all the 82576 requests (read/write, snoop and no snoop).

If a request can be sent as a single PCIe packet and still meet rules 1-3, then it is not broken at a cache-line boundary (as defined in the PCIe Cache line size configuration word), but rather, sent as a single packet (motivation is that the chipset might break the request along cache-line boundaries, but the 82576 should still benefit from better PCIe utilization). However, if rules 1-3 require that the request is broken into two or more packets, then the request is broken at a cache-line boundary.

3.1.4.2.2 Multiple Tx Data Read Requests

The 82576 supports four pipe lined requests for transmit data. In general, the four requests might belong to the same packet or to consecutive packets. However, the following restriction applies:

- All requests for a packet are issued before a request is issued for a consecutive packet

Read requests can be issued from any of the supported queues, as long as the restriction is met. Pipelined requests might belong to the same queue or to separate queues. However, as previously noted, all requests for a certain packet are 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 82576 handles completions that arrive in any order. Once all completions arrive for a given request, the 82576 might issue the next pending read data request.

- The 82576 incorporates a 2 KB re-order 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 the minimum {512, Max_Read_Request_Size}.

In addition to the four pipeline requests for transmit data, the 82576 can issue a single read request for each of the Tx descriptors and Rx descriptors. The requests for 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).

3.1.4.3 Messages

3.1.4.3.1 Message Handling by the 82576 (as a Receiver)

Message packets are special packets that carry a message code.

The upstream device transmits special messages to the 82576 by using this mechanism.

The transaction layer decodes the message code and responds to the message accordingly.

Table 3-6. Supported Message in the 82576 (as a Receiver)

Message code [7:0]	Routing r2r1r0	Message	Device later response
0x14	100	PM_Active_State_NAK	Internal signal set
0x19	011	PME_Turn_Off	Internal signal set
0x50	100	Slot power limit support (has one Dword data)	Silently drop

Table 3-6. Supported Message in the 82576 (as a Receiver) (Continued)

0x7E	010,011,100	Vendor_defined type 0 no data	Unsupported request[1]
0x7E	010,011,100	Vendor_defined type 0 data	Unsupported request[1]
0x7F	010,011,100	Vendor_defined type 1 no data	Silently drop
0x7F	010,011,100	Vendor_defined type 1 data	Silently drop
0x00	011	Unlock	Silently drop

[1] No Completion is expected for this type of packets

3.1.4.3.2 Message Handling by the 82576 (as a Transmitter)

The transaction layer is also responsible for transmitting specific messages to report internal/external events (such as interrupts and PMEs).

Table 3-7. Supported Message in the 82576 (as a Transmitter)

Message code [7:0]	Routing r2r1r0	Message
0x20	100	Assert INT A
0x21	100	Assert INT B
0x22	100	Assert INT C
0x23	100	Assert INT D
0x24	100	De-assert INT A
0x25	100	De-assert INT B
0x26	100	De-assert INT C
0x27	100	De-Assert INT D
0x30	000	ERR_COR
0x31	000	ERR_NONFATAL
0x33	000	ERR_FATAL
0x18	000	PM_PME
0x1B	101	PME_TO_ACK

3.1.4.4 Ordering Rules

The 82576 meets the PCIe ordering rules (PCI-X rules) by following the PCI simple device model:

- Deadlock avoidance - Master and target accesses are independent - The response to a target access does not depend on the status of a master request to the bus. If master requests are blocked, such as due to no credits, target completions might still proceed (if credits are available).
- Descriptor/data ordering - The 82576 does not proceed with some internal actions until respective data writes have ended on the PCIe link:
  - — The 82576 does not update an internal header pointer until the descriptors that the header pointer relates to are written to the PCIe link.
  - — The 82576 does not issue a descriptor write until the data that the descriptor relates to is written to the PCIe link.

The 82576 might issue the following master read request from each of the following clients:

- Rx Descriptor Read (one for each LAN port)
- Tx Descriptor Read (two for each LAN port)
- Tx Data Read (up to four for each LAN port/ one for the manageability)

Completion separate read requests are not guaranteed to return in order. Completions for a single read request are guaranteed to return in address order.

3.1.4.4.1 Out of Order Completion Handling

In a split transaction protocol, when using multiple read requests in a multi processor environment, there is a risk that completions arrive from the host memory out of order and interleaved. In this case, the 82576 sorts the request completion and transfers them to the Ethernet in the correct order.

3.1.4.5 Transaction Definition and Attributes

3.1.4.5.1 Max Payload Size

The 82576 policy to determine Max Payload Size (MPS) is as follows:

- Master requests initiated by the 82576 (including completions) limits MPS to the value defined for the function issuing the request.
- Target write accesses to the 82576 are accepted only with a size of one Dword or two Dwords. Write accesses in the range of (three Dwords, MPS, etc.) are flagged as UR. Write accesses above MPS are flagged as malformed.

3.1.4.5.2 Traffic Class (TC) and Virtual Channels (VC)

The 82576 only supports TC=0 and VC=0 (default).

3.1.4.5.3 Relaxed Ordering

The 82576 takes advantage of the relaxed ordering rules in PCIe. By setting the relaxed ordering bit in the packet header, the 82576 enables the system to optimize performance in the following cases:

- Relaxed ordering for descriptor and data reads: When the 82576 emits a read transaction, its split completion has no ordering 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 82576 masters receive data writes, it also enables them to bypass each other in the path to system memory because software does not process this data until their associated descriptor writes complete.
- The 82576 cannot relax ordering for descriptor writes, MSI/MSI-X writes or PCIe messages.

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

Relaxed ordering is enabled in the 82576 by clearing the *RO_DIS* bit in the CTRL_EXT register. Actual setting of relaxed ordering is done for LAN traffic by the host through the DCA registers.

3.1.4.5.4 Snoop Not Required

The 82576 sets the *Snoop Not Required* attribute bit for master data writes. System logic might provide a separate path into system memory for non-coherent traffic. The non-coherent path to system memory provides 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 set the relaxed ordering attribute as well (assuming that system logic supports both attributes). In fact, some chipsets require that relaxed ordering is set for no-snoop to take effect.

Global no-snoop support is enabled in the 82576 by clearing the *NS_DIS* bit in the CTRL_EXT register. Actual setting of no snoop is done for LAN traffic by the host through the DCA registers.

3.1.4.5.5 No Snoop and Relaxed Ordering for LAN Traffic

Software might configure non-snoop and relax order attributes for each queue and each type of transaction by setting the respective bits in the RXCTRL and TXCTRL registers.

Table 3-8 lists the default behavior for the *No-Snoop* and *Relaxed Ordering* bits for LAN traffic when I/OAT 2 is enabled.

Table 3-8. LAN Traffic Attributes

Transaction	No-Snoop Default	Relaxed Ordering Default	Comments
Rx Descriptor Read	N	Y	
Rx Descriptor Write-Back	N	N	Relaxed ordering must never be used for this traffic.
Rx Data Write	Y	Y	See the following note and Section 3.1.4.5.5.1
Rx Replicated Header	N	Y	
Tx Descriptor Read	N	Y	
Tx Descriptor Write-Back	N	Y	
Tx TSO Header Read	N	Y	
Tx Data Read	N	Y	

Note: Rx payload no-snoop is also conditioned by the *NSE* bit in the receive descriptor. See Section 3.1.4.5.5.1.

3.1.4.5.5.1 No-Snoop Option for Payload

Under certain conditions, which occur when I/OAT is enabled, software knows that it is safe to transfer (DMA) a new packet into a certain buffer without snooping on the front-side bus. This scenario typically occurs when software is posting a receive buffer to hardware that the CPU has not accessed since the last time it was owned by hardware. This might happen if the data was transferred to an application buffer by the I/OAT DMA engine.

In this case, software should be able to set a bit in the receive descriptor indicating that the 82576 should perform a no-snoop DMA transfer when it eventually writes a packet to this buffer.

When a non-snoop transaction is activated, the TLP header has a non-snoop attribute in the *Transaction Descriptor* field.

This is triggered by the *NSE* bit in the receive descriptor. See Section 7.1.5.

3.1.4.5.5.2 No Snoop Option for TSO Header

As hardware reads the header of a TSO request for each segment it sends, we may safely assume that after the first read of the header it is updated in the main memory. As as result, all the subsequent reads of the header might be done with the no-snoop option set. This option is triggered by setting the *NoSnoop_LSO_hdr_buf* bit in the DTXCTL register.

3.1.4.6 Flow Control

3.1.4.6.1 82576 Flow Control Rules

The 82576 implements only the default Virtual Channel (VC0). A single set of credits is maintained for VC0.

Table 3-9. Allocation of FC Credits

Credit Type	Operations	Number Of Credits
Posted Request Header (PH)	Target Write (one unit) Message (one unit)	Two units (to enable concurrent accesses to both LAN ports).
Posted Request Data (PD)	Target Write (Length/16 bytes=1) Message (one unit)	MAX_PAYLOAD_SIZE/16
Non-Posted Request Header (NPH)	Target Read (one unit) Configuration Read (one unit) Configuration Write (one unit)	Two units (to enable concurrent target accesses to both LAN ports).
Non-Posted Request Data (NPD)	Configuration Write (one unit)	Two units.
Completion Header (CPLH)	Read Completion (N/A)	Infinite (accepted immediately).
Completion Data (CPLD)	Read Completion (N/A)	Infinite (accepted immediately).

Rules for FC updates:

- The 82576 maintains two credits for NPD at any given time. It 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 82576 provides two credits for PH (such as for two concurrent target writes) and two credits for NPH (such as for two concurrent target reads). UpdateFC packets are scheduled immediately after a resource becomes available.
- The 82576 follows the PCIe recommendations for frequency of UpdateFC FCPs.

3.1.4.6.2 Upstream Flow Control Tracking

The 82576 issues a master transaction only when the required FC credits are available. Credits are tracked for posted, non-posted, and completions (the later to operate against a switch).

3.1.4.6.3 Flow Control Update Frequency

In any case, UpdateFC packets are scheduled immediately after a resource becomes available.

When the link is in the L0 or L0s link state, Update FCPs for each enabled type of non-infinite FC credit must be scheduled for transmission at least once every 30 µs (-0%/+50%), except when the *Extended Sync* bit of the Control Link register is set, in which case the limit is 120 µs (-0%/+50%).

3.1.4.6.4 Flow Control Timeout Mechanism

The 82576 implements the optional FC update timeout mechanism.

The mechanism is activated when the Link is in L0 or L0s Link state. It uses a timer with a limit of 200µs (-0%/+50%), where the timer is reset by the receipt of any Init or Update FCP. Alternately, the timer may be reset by the receipt of any DLLP.

After timer expiration, the mechanism instructs the PHY to re-establish the link (via the LTSSM recovery state).

3.1.4.7 Error Forwarding

If a TLP is received with an error-forwarding trailer, the packet is dropped and not delivered to its destination. The 82576 does not initiate any additional master requests for that PCI function until it detects an internal reset or a software reset for the associated LAN. Software is able to access device registers after such a fault.

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

3.1.5 Data Link Layer

3.1.5.1 ACK/NAK Scheme

The 82576 supports two alternative schemes for ACK/NAK rate:

1. ACK/NAK is scheduled for transmission according to timeouts specified in the LTIV register
2. ACK/NAK is scheduled for transmission according to timeouts specified in the PCIe specification.

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

3.1.5.2 Supported DLLPs

The following DLLPs are supported by the 82576 as a receiver:

Table 3-10. DLLPs Received by the 82576

DLLP type	Remarks
Ack	
Nak	
PM_Request_Ack	
InitFC1-P	Virtual Channel 0 only
InitFC1-NP	Virtual Channel 0 only
InitFC1-Cpl	Virtual Channel 0 only
InitFC2-P	Virtual Channel 0 only
InitFC2-NP	Virtual Channel 0 only
InitFC2-Cpl	Virtual Channel 0 only
UpdateFC-P	Virtual Channel 0 only
UpdateFC-NP	Virtual Channel 0 only
UpdateFC-Cpl	Virtual Channel 0 only

The following DLLPs are supported by the 82576 as a transmitter:

Table 3-11. DLLPs Initiated by the 82576

DLLP type	Remarks
Ack	
Nak	
PM_Enter_L1	
PM_Enter_L23	

Table 3-11. DLLPs Initiated by the 82576 (Continued)

PM_Active_State_Request_L1	
InitFC1-P	Virtual Channel 0 only
InitFC1-NP	Virtual Channel 0 only
InitFC1-Cpl	Virtual Channel 0 only
InitFC2-P	Virtual Channel 0 only
InitFC2-NP	Virtual Channel 0 only
InitFC2-Cpl	Virtual Channel 0 only
UpdateFC-P	Virtual Channel 0 only
UpdateFC-NP	Virtual Channel 0 only

Note: UpdateFC-Cpl is not sent because of the infinite FC-Cpl allocation.

3.1.5.3 Transmit EDB Nullifying

In case of a retrain necessity, there is a need to guarantee that no abrupt termination of the Tx packet happens. For this reason, early termination of the transmitted packet is possible. This is done by appending an EDB (EnD Bad symbol) to the packet.

3.1.6 Physical Layer

3.1.6.1 Link Width

The 82576 supports a maximum link width of x4, x2, or x1 as determined by the Lane_Width field in PCIe Init Configuration 3 EEPROM word.

The max link width is loaded into the *Maximum Link Width* field of the PCIe Capability register (LCAP[11:6]). The hardware default is x4 link.

During link configuration, the platform and the 82576 negotiate on a common link width. The link width must be one of the supported PCIe link widths (x1, x2, x4), such that:

- If *Maximum Link Width* = x4, then the 82576 negotiates to either x4, x2 or x1.[1]
- If *Maximum Link Width* = x2, then the 82576 negotiates to either x2 or x1.
- If *Maximum Link Width* = x1, then the 82576 only negotiates to x1.

3.1.6.2 Polarity Inversion

If polarity inversion is detected, the receiver must invert the received data.

During the training sequence, the receiver looks at Symbols 6-15 of TS1 and TS2 as the indicator of lane polarity inversion (D+ and D- are swapped). If lane polarity inversion occurs, the TS1 Symbols 6-15 received are D21.5 as opposed to the expected D10.2. Similarly, if lane polarity inversion occurs, Symbols 6-15 of the TS2 ordered set are D26.5 as opposed to the expected D5.2. This provides the clear indication of lane polarity inversion.

3.1.6.3 L0s Exit latency

The number of FTS sequences (N_FTS) sent during L1 exit, is loaded from the EEPROM into an 8-bit read-only register.

[1] See restriction in Section 3.1.6.5.

3.1.6.4 Lane-to-Lane De-Skew

A multi-lane link might have many sources of lane to lane skew. Although symbols are transmitted simultaneously on all lanes, they cannot be expected to arrive at the receiver without lane-to-lane skew. The lane-to-lane skew can include components, which are less than a bit time, bit time units (400 ps for 2.5 Gb), or full symbol time units (4 ns) of skew caused by the retiming repeaters' insert/ delete operations. Receivers use TS1 or TS2 or Skip Ordered Sets (SOS) to perform link de-skew functions.

The 82576 supports de-skew of up to 6 symbols time (24 ns).

3.1.6.5 Lane Reversal

The following lane reversal modes are supported (see Figure 3-2):

- Lane configuration of x4, x2, and x1
- Lane reversal in x4 and in x2
- Degraded mode (downshift) from x4 to x2 to x1 and from x2 to x1, with one restriction - if lane reversal is executed in x4, then downshift is only to x1 and not to x2.

Note: The restriction requires that a x2 interface to the 82576 must connect to lanes 0 and 1 on the 82576. The PCIe Card Electromechanical specification does not allow to route a x2 link to a wider connector. Therefore, a system designer is not allowed to connect a x2 link to lanes 2 and 3 of a PCIe connector. It is also recommended that when used in x2 mode on a NIC, the 82576 is connected to lanes 0 and 1 of the NIC.

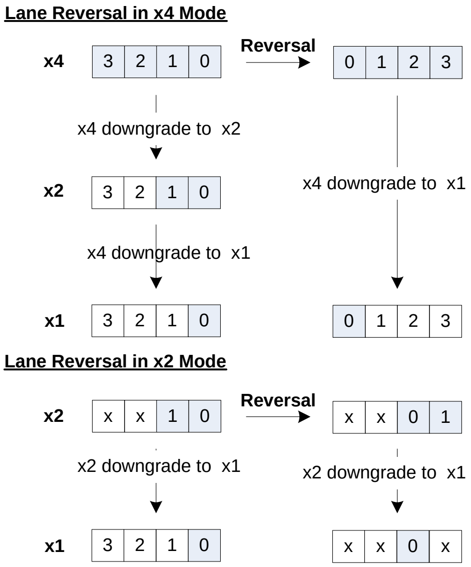


Figure 3-2. Lane Reversal Supported Modes

Configuration bits:

- EEPROM *Lane Reversal Disable* bit - disables lane reversal altogether

3.1.6.6 Reset

The PCIe PHY can supply core reset to the 82576. The reset can be caused by two sources:

1. Upstream move to hot reset - Inband Mechanism (LTSSM).
2. Recovery failure (LTSSM returns to detect).
3. Upstream component moves to Disable.

3.1.6.7 Scrambler Disable

The scrambler/de-scrambler functionality in the 82576 can be eliminated by two mechanisms:

1. Upstream according to the PCIe specification.
2. EPROM bit.

3.1.7 Error Events and Error Reporting

3.1.7.1 Mechanism in General

PCIe defines two error reporting paradigms: the baseline capability and the Advanced Error Reporting (AER) capability. The baseline error reporting capabilities are required of all PCIe devices and define the minimum error reporting requirements. The AER capability is defined for more robust error reporting and is implemented with a specific PCIe capability structure.

Both mechanisms are supported by the 82576.

Also the *SERR# Enable* and the *Parity Error* bits from the legacy Command register take part in the error reporting and logging mechanism.

Figure 3-3 shows, in detail, the flow of error reporting in the 82576.

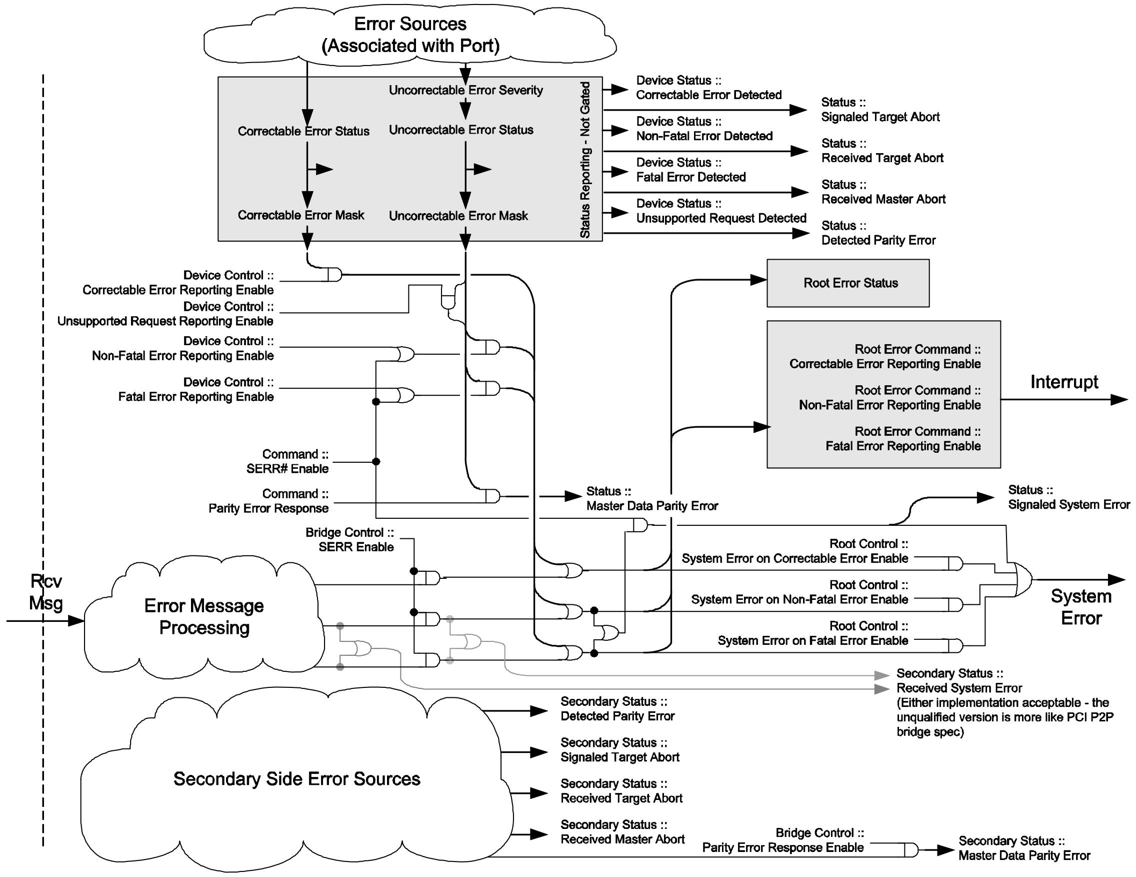


Figure 3-3. Error Reporting Mechanism

3.1.7.2 Error Events

Table 3-12 lists the error events identified by the 82576 and the response in terms of logging, reporting, and actions taken. Consult the PCIe specification for the effect on the PCI Status register.

Table 3-12. Response and Reporting of Error Events

Error Name	Error Events	Default Severity	Action
PHY errors			
Receiver error	8b/10b decode errors Packet framing error	Correctable. Send ERR_CORR	TLP to initiate NAK and drop data. DLLP to drop.

Table 3-12. Response and Reporting of Error Events (Continued)

Data link errors			
Bad TLP	• Bad CRC • Not legal EDB • Wrong sequence number	Correctable. Send ERR_CORR	TLP to initiate NAK and drop data.
Bad DLLP	• Bad CRC	Correctable. Send ERR_CORR	DLLP to drop.
Replay timer timeout	• REPLAY_TIMER expiration	Correctable. Send ERR_CORR	Follow LL rules.
REPLAY NUM rollover	• REPLAY NUM rollover	Correctable. Send ERR_CORR	Follow LL rules.
Data link layer protocol error	• Received ACK/NACK not corresponding to any TLP	Uncorrectable. Send ERR_FATAL	Follow LL rules.
TLP errors			
Poisoned TLP received	• TLP with error forwarding	Uncorrectable. ERR_NONFATAL Log header	A poisoned completion is ignored and the request can be retried after timeout. If enabled, the error is reported.
Unsupported Request (UR)	• Wrong config access • MRdLk • Configuration request type 1 • Unsupported vendor Defined type 0 message • Not valid MSG code • Not supported TLP type • Wrong function number • Wrong TC/VC • Received target access with data size > 64-bit • 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	• Attempts to write to the Flash device when writes are disabled (EEC.FWE=10b)	Uncorrectable. ERR_NONFATAL Log header	Send completion with CA.
Unexpected completion	• Received completion without 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. The 82576 doesn’t report violations of Flow Control initialization protocol

Table 3-12. Response and Reporting of Error Events (Continued)

Malformed TLP (MP)	• Data payload exceed Max_Payload_Size • Received TLP data size does not match length field • TD field value does not correspond with the observed size • Byte enables violations. • Power management messages that don't use TC0. • Usage of unsupported VC	Uncorrectable. ERR_FATAL Log header	Drop the packet and free FC credits.
Completion with unsuccessful completion status		No action (already done by originator of completion).	Free FC credits.
Byte count integrity in completion process.	When 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 82576 doesn't check for this error and accepts these packets. This may cause a completion timeout condition.

3.1.7.3 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 that occur for the same packet are not signaled at the transaction layer.

3.1.7.4 Completion with Unsuccessful Completion Status

A completion with 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 data that was not delivered.

3.1.7.5 Error Reporting Changes

The Rev. 1.1 specification defines two changes to advanced error reporting. A new *Role-Based Error Reporting* bit in the Device Capabilities register is set to 1b to indicate that these changes are supported by the 82576.

1. Setting the *SERR# Enable* bit in the PCI Command register also enables UR reporting (in the same manner that the *SERR# Enable* bit enables reporting of correctable and uncorrectable errors). In other words, the *SERR# Enable* bit overrides the UR *Error Reporting Enable* bit in the PCIe Device Control register.
2. Changes in the response to some uncorrectable non-fatal errors, detected in non-posted requests to the 82576. These are called advisory Non-fatal error cases. For each of the errors that follow, the following behavior is defined:
   a. The *Advisory Non-Fatal Error Status* bit is set in the Correctable Error Status register to indicate the occurrence of the advisory error and the *Advisory Non-Fatal Error Mask* corresponding bit in the Correctable Error Mask register is checked to determine whether to proceed further with logging and signaling.

b. If the *Advisory Non-Fatal Error Mask* bit is clear, logging proceeds by setting the corresponding bit in the Uncorrectable Error Status register, based upon the specific uncorrectable error that's being reported as an advisory error. If the corresponding uncorrectable error bit in the Uncorrectable Error Mask register is clear, the First Error Pointer and Header Log registers are updated to log the error, assuming they are not still occupied by a previously unserviced error.

c. An ERR_COR message is sent if the *Correctable Error Reporting Enable* bit is set in the Device Control register. An ERROR_NONFATAL message is not sent for this error.

The following uncorrectable non-fatal errors are considered as advisory non-fatal Errors:

- A completion with an Unsupported Request or Completer Abort (UR/CA) status that signals an uncorrectable error for a non-posted request. If the severity of the UR/CA error is non-fatal, the completer must handle this case as an advisory non-fatal error.
- When the requester of a non-posted request times out while waiting for the associated completion, the requester is permitted to attempt to recover from the error by issuing a separate subsequent request, or to signal the error without attempting recovery. The requester is permitted to attempt recovery zero, one, or multiple (finite) times, but must signal the error (if enabled) with an uncorrectable error message if no further recovery attempt is made. If the severity of the completion timeout is non-fatal and the requester elects to attempt recovery by issuing a new request, the requester must first handle the current error case as an advisory non-fatal error.
- When a receiver receives an unexpected completion and the severity of the unexpected completion error is non-fatal, the receiver must handle this case as an advisory non-fatal error.

3.1.8 Performance Monitoring

The 82576 incorporates PCIe performance monitoring counters to provide common capabilities to evaluate performance. The 82576 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. software can reset, stop, or start the counters (all at the same time).

The list of events supported by the 82576 and the counters control bits are described in the memory register map (Section 8.6).

Some counters operate with a threshold - the counter increments only when the monitored event crossed a configurable threshold (such as the number of available credits is below a threshold).

Counters operate in the following modes:

- Count mode - The counter increments when the respective event occurred.
- Leaky bucket mode - The counter increments only when the rate of events exceeded a certain value. See Section 3.1.8.1.

3.1.8.1 Leaky Bucket Mode

Each of the counters may be configured independently to operate in a leaky bucket mode. When in leaky bucket mode, the following functionality is provided:

- One of four 16-bit Leaky Bucket Counters (LBC) is enabled via the *LBC Enable [3:0]* bits in the PCIe Statistic Control register #1.
- The LBC is controlled by the *GIO_COUNT_START*, *GIO_COUNT_STOP*, and *GIO_COUNT_RESET* bits in the PCIe Statistic Control register #1.
- The LBC increments every time the respective event occurs.
- The LBC is decremented every T ms as defined in the *LBC Timer* field in the PCIe Statistic Control registers.
- When an event occurs and the value of the LBC meets or exceeds the threshold defined in the *LBC Threshold* field in the PCIe Statistic Control registers, the respective statistics counter increments.

3.1.9 PCIe Power Management

Described in Section 5.4.1 - Power Management.

3.1.10 PCIe Programming Interface

Described in Section 9. - PCIe Programming Interface

3.2 Management Interfaces

The 82576 contains 2 possible interfaces to an external BMC.

- SMBus
- NC-SI

Since the manageability sideband throughput is lower than the network link throughput, the 82576 allocates an 8 KB internal buffer for incoming network packets prior to being sent over the sideband interface.

3.2.1 SMBus

SMBus is an optional interface for pass-through and/or configuration traffic between an external BMC and the 82576. The SMBus commands used to configure or read status from the 82576 are described in Chapter 10..

3.2.1.1 Channel Behavior

3.2.1.1.1 SMBus Addressing

The SMBus addresses that the 82576 responds to depend on the LAN mode (teaming/non-teaming). When the LAN is in teaming mode (fail-over), the 82576 is presented over the SMBus as one device along with one SMBus address. When in non-teaming mode in the LAN ports, the SMBus is presented as two SMBus devices on the SMBus along with two SMBus addresses. In dual-address mode all pass-through functionality is duplicated on the SMBus address, where each SMBus address is connected to a different LAN port.

Note: DO NOT configure both ports to the same address. When a LAN function is disabled, the corresponding SMBus address is not presented to the external BMC.

The SMBus address method is defined through the *SMBus Addressing Mode* bit in the EEPROM. The SMBus addresses are set by *SMBus Address 0* and *SMBus Address 1* in the EEPROM.

Note: If the single-address mode is set, only *SMBus address 0* field is valid.

The SMBus addresses (those that are enabled from the EEPROM) can be re-assigned using the SMBus ARP protocol.

Besides the SMBus address values, all the previously stated parameters of the SMBus (SMBus channel selection, address mode, address enable) can be set only through EEPROM configuration. The EEPROM is read on the 82576 at power-up, resets, and other cases described in Section 4.2.

All SMBus addresses should be in Network Byte Order (NBO); most significant byte first.

3.2.1.1.2 SMBus Notification Methods

The 82576 supports three methods of informing the external BMC that it has information that is needed to be read by an external BMC:

- SMBus alert.
- Asynchronous notify.
- Direct receive.

The notification method that is used by the 82576 can be configured from the SMBus using the Receive Enable command. The default method is set from the EEPROM in the *PT init* field.

The following events cause the 82576 to send a notification event to the external BMC:

- Receiving a LAN packet that was designated to the BMC.
- Receiving a request status command from the BMC initiates a status response (see Section 10.3.2.1.2).
- Status change has occurred and the 82576 is configured to notify the external BMC upon one of the status changes. The following event triggers a notification to the BMC:
  - — A change in any of the *Status Data 1* bits of the Read Status command (see Section 10.3.2.2.3 for description of this command).

There might be cases where the external BMC is hung and is unable to respond to the SMBus notification. The 82576 has a time-out value defined in the EEPROM (see Section 6.8) to avoid hanging while waiting for the notification response. If the BMC does not respond until the timeout expires, the notification is de-asserted.

3.2.1.1.2.1 SMBus Alert and Alert Response Method

The SMBus Alert# signal is an additional SMBus signal that acts as an asynchronous interrupt signal to an external SMBus master. The 82576 asserts this signal each time it has a message that it needs the external BMC to read and if the chosen notification method is the SMBus-alert method. Note that the SMBus alert is an open-drain signal, which means that other devices besides the 82576 can be connected on the same alert pin and the external BMC needs a mechanism to distinguish between the alert sources as described:

The external BMC can respond to the alert by issuing an ARA cycle (see Figure 3-13) to detect the alert source device. The 82576 responds to the ARA cycle (if it was the SMBus alert source) and de-asserts the alert when the ARA cycle completes. Following the ARA cycle, the external BMC issues a Read command to retrieve the 82576 message.

Some BMCs do not implement ARA cycle transactions. These BMCs respond to an alert by issuing a Read command to the 82576 (0xC0/0xD0 or 0xDE). The 82576 always responds to a Read command, even if it is not the source of the notification. The default response is a status transaction. If the 82576 is the source of the SMBus alert, it replies to the read transaction and de-asserts the alert after the command byte of the read transaction.

The ARA cycle is an SMBus receive byte transaction to SMBus Address 0001-100b. Note that the ARA transaction does not support PEC. The ARA transaction format is as follows:

Table 3-13. SMBus ARA Cycle Format

1	7	1	1	8	1	1
S	Alert Response Address	Rd	A	Slave Device Address	A	P
	0001 100	1	0	Manageability Slave SMBus Address	1	

Note: Since the master-receiver (BMC receiver) is involved in the transaction, it must signal the end of data by generating a NACK (a ‘1’ in the ACK bit position) on the slave device address byte that was clocked out. This releases the data line to allow the master to generate a stop condition.

3.2.1.1.2.2 Asynchronous Notify Method

When configured to asynchronous notify method, the 82576 acts as SMBus master and notifies the external BMC by issuing a modified form of the write word transaction. The asynchronous notify transaction SMBus address and data payload is configured using the Receive Enable command or using the EEPROM defaults. Note that the asynchronous notify method is not protected by a PEC byte.

Table 3-14. Asynchronous Notify Command Format

1	7	1	1	7	1	1	
S	Target Address	Wr	A	Sending Device Address		A	• • •
	BMC Slave Address	0	0	Manageability Slave SMBus Address	0	0	

8	1	8	1	1
Data Byte Low	A	Data Byte High	A	P
Interface	0	Alert Value	0	

The target address and data byte low/high is taken from the Receive Enable command (see Section 10.3.2.1.3) or EEPROM configuration (See Section 6.8).

3.2.1.1.2.3 Direct Receive Method

If configured, the 82576 has the capability to send the message it needs to transfer to the external BMC as a master over the SMBus, instead of alerting the BMC, and waiting for it to read the message.

Table 3-15 shows the message. Note that the “F”, “L” and command fields in the message are the same as the op-code returned by the 82576 in response to a BMC Receive TCO Packet Block read command (See Section 10.3.2.2.1). The rules for the “F” and “L” flags are also the same as used in the Receive TCO Packet Block Read command.

Table 3-15. Direct Receive Transaction Format

1	7	1	1	1	1	6	1	
S	Target Address	Wr	A	F	L	Command	A	• • •
	BMC Slave Address	0	0	First Flag	Last Flag	Receive TCO Command 01 0000b	0	

8	1	8	1		1	8	1	1
Byte Count	A	Data Byte 1	A	• • •	A	Data Byte N	A	P
N	0		0		0		0	

3.2.1.1.3 Receive TCO Flow

The 82576 is used as a channel for receiving packets from the network link and passing them to the external BMC. The BMC can configure the 82576 to pass specific packets to the BMC as described in Section 10.3.2.1.6. Once a full packet is received from the link and identified as a manageability packet that should be transferred to the BMC, the 82576 starts the receive TCO transaction flow to the BMC.

The maximum SMBus fragment length is defined in the EERPOM (See Section 6.8.2). The 82576 uses the SMBus notification method to notify the BMC that it has data to deliver. The packet is divided into fragments, where the 82576 uses the maximum fragment size allowed in each fragment. The last fragment of the packet transfer is always the status of the packet (see Section 10.3.2.2.2). As a result, the packet is transferred in at least two fragments. The data of the packet is transferred in the Receive TCO LAN packet transaction as described in Section 10.3.2.2.1.

When SMBus alert is selected as the BMC notification method, the 82576 notifies the BMC on each fragment of a multi-fragment packet. When asynchronous notify is selected as the BMC notification method, the 82576 notifies the BMC only on the first fragment of a received packet. It is BMC's responsibility to read the full packet including all the fragments.

Any timeout on the SMBus notification results in discarding the entire packet. Any NACK by the BMC on one of the 82576 receive bytes also causes the packet to be silently discarded.

The maximum size of the received packet is limited by the 82576 hardware to 1536 bytes. Packets larger then 1536 bytes are silently discarded. Any packet smaller than 1536 bytes is processed by the 82576.

Note: When the *RCV_EN* bit is cleared (see Section 10.3.2.1.3), all receive TCO functionality is disabled, not just the packets that are directed to the BMC (also auto ARP packets).

3.2.1.1.4 Transmit TCO Flow

The 82576 is used as a channel for transmitting packets from the external BMC to the network link. The network packet is transferred from the external BMC over the SMBus, and then, when fully received by the 82576, is transmitted over the network link.

In dual-address mode, each SMBus address is connected to a different LAN port. When a packet is received in SMBus transactions using *SMBus Address 0*, it is transmitted to the network using LAN port 0 and is transmitted through LAN port 1, if received on *SMBus Address 1*. In single-address mode, the transmitted port is selected according to the fail-over algorithm (see Section 3.2.1.1.9).

The 82576 supports packets up to the Ethernet packet length (1536 bytes). SMBus transactions can be up to 240 bytes in length, which means that packets can be transferred over the SMBus in more than one fragment. In each command byte there are the *F* and *L* bits. When the *F* bit is set, it means that this is the first fragment of the packet; *L* means that it is the last fragment of the packet.

Note: When both flags are set, the entire packet is in one fragment.

The packet is sent over the network link, only after all its fragments are received correctly over the SMBus.

The 82576 calculates the L2 CRC on the transmitted packet and adds its four bytes at the end of the packet. Any other packet field (such as XSUM) must be calculated and inserted by the external BMC (the 82576 does not change any field in the transmitted packet, besides adding padding and CRC bytes).

Note: If the packet sent by the BMC is larger than 1536 bytes, then the packet is silently discard by the 82576.

The minimum packet length defined by the 802.3 specification is 64 bytes. The 82576 pads packets that are less than 64 bytes to meet the specification requirements. There is one exception, when the packet sent over the SMBus is less than 32 bytes, the external BMC must pad it for at least 32 bytes. The passing bytes value should be zero.

Note: Packets that are smaller then 32 bytes (including padding) are silently discarded by the 82576.

If the network link goes down at anytime while the 82576 is receiving the packet, it silently discards the packet. Note that any link down event during the transfer of a packet over the SMBus (after received from the network), does not stop the operation.

The transmit SMBus transactions are described in Section 10.3.2.1.

3.2.1.1.5 Transmit Errors in Sequence Handling

Once a packet is transferred over the SMBus from the BMC to the 82576, the *F* and *L* flags should follow specific rules. The *F* flag defines that this is the first fragment of the packet; The *L* flag defines that the transaction contains the last fragment of the packet.

The following table lists the different options regarding the flags in transmit packet transactions:

Table 3-16. Flags in Transmit Packet Transactions

Previous	Current	Action / Notes
Last	First	Accepts both.
Last	Not First	Error for current transaction. Current transaction is discarded and an abort status is asserted.
Not Last	First	Error for previous transaction. Previous transaction (until previous first) is discarded. Current packet is processed. No abort status is asserted.
Not Last	Not First	Processes the current transaction.

Note that since every other Block Write command in the TCO protocol has both *F* and *L* flags off, they cause flushing any pending transmit fragments that were previously received. In other words, when running the TCO transmit flow, no other block write transactions are allowed in between the fragments.

3.2.1.1.6 TCO Command Aborted Flow

Bit 6 in first byte of the status returned from the 82576 to the external BMC indicates that there was a problem with previous SMBus transactions or with the completion of the operation requested in previous transaction.

An abort can be asserted for any of the following reasons:

- Any error in the SMBus protocol (NACK, SMBus timeouts).
- Any error in compatibility between required protocols to specific functionality (Receive Enable command with byte count not 1/14 as defined in the command specification).
- If the 82576 does not have space to store the transmit packet from the BMC (in its internal buffer before sending it to the link). In this case, the entire transaction completes, but the packet is discarded and the BMC is notified about it through the *Abort* bit.
- Error in the *F*/*L* bit sequence during multi-fragment transactions.
- The *Abort* bit is asserted after an internal reset to the 82576 manageability unit.

Note: An abort in the status does not always imply that the last transaction of the sequence was incorrect. There is a gap between the time the status is read from the 82576 and the time the transaction occurred.

3.2.1.1.7 Concurrent SMBus Transactions

In single-address mode concurrent SMBus transactions (receive, transmit and configuration read/write) are allowed. Transmit fragments can be sent between receive fragments and configuration Read/Write commands can also be issued between receive and transmit fragments.

In dual-address mode the same rules also apply for concurrent traffic between the two addresses supported by the 82576.

3.2.1.1.8 SMBus ARP Functionality

The 82576 supports SMBus ARP protocol as defined in the SMBus 2.0 specification. The 82576 is a persistent slave address device meaning that its SMBus address is valid after power-up and loaded from the EEPROM. The 82576 supports all SMBus ARP commands defined in the SMBus specification, both general and directed.

Note: SMBus ARP can be disabled through EEPROM configuration (See Section 6.8.3).

SMBus-ARP transactions are described in Section 10.3.2.3.

3.2.1.1.8.1 SMBus ARP in Dual-/Single-Address Mode

The 82576 operates either in single SMBus address mode or in dual SMBus address mode. These modes reflect on its SMBus-ARP behavior.

When operating in single-address mode, the 82576 presents itself on the SMBus as one device and responds to SMBus-ARP as one device only. In this case, its SMBus address is *SMBus Address 0* as defined in the EEPROM SMBus ARP addresses word (see Section 6.7.31 and Section 6.7.32). The 82576 has only one *AR* flag and one *AV* flag. The vendor specific ID, which is the MAC address of the LAN's port, is taken from the port 0 address.

In dual-address mode, the 82576 responds as two SMBus devices, meaning that it has two sets of *AR*/*AV* flags (one for each port). The 82576 responds twice to the SMBus-ARP master, one time for each port. Both SMBus addresses are taken from the SMBus ARP addresses word of the EEPROM. The UDID is different between the two ports in the vendor specific ID field, which represent the MAC address, which is different between the two ports. It is recommended for the 82576 to first answer as port 0, and only when the address is assigned, to start answering as port 1 to the Get UDID command.

3.2.1.1.8.2 SMBus ARP Flow

SMBus-ARP flow is based on the status of two flags:

- *AV* - Address Valid - This flag is set when the 82576 has a valid SMBus address.
- *AR* - Address Resolved - This flag is set when the 82576’s SMBus address is resolved (SMBus address was assigned by the SMBus-ARP process).

Note: These flags are internal the 82576 flags and not shown to external SMBus devices.

Since the 82576 is a Persistent SMBus Address (PSA) device, the *AV* flag is always set, while the *AR* flag is cleared after power-up until the SMBus-ARP process completes. Since the *AV* flag is always set, the 82576 always has a valid SMBus address.

When the SMBus master needs to start an SMBus-ARP process, it resets (In terms of ARP functionality) all the devices on the SMBus by issuing either Prepare to ARP or Reset Device commands. When the 82576 accepts one of these commands, it clears its *AR* flag (if set from previous SMBus-ARP process), but not its *AV* flag (the current SMBus address remains valid until the end of the SMBus ARP process).

The meaning of an *AR* flag cleared is that the 82576 answers the following SMBus ARP transactions that are issued by the master. The SMBus master then issues a Get UDID command (general or directed), to identify the devices on the SMBus. The 82576 responds to the directed command all the time and to the general command only if its *AR* flag is not set. After the Get UDID command, the master assigns the 82576's SMBus address by issuing an Assign Address command. The 82576 checks whether the UDID matches its own UDID, and if they match, it switches its SMBus address to the address assigned by the command (byte 17). After accepting the Assign Address command, the *AR* flag is set and from this point on (as long as the *AR* flag is set), the 82576 does not respond to the Get UDID general command, while all other commands should be processed even if the *AR* flag is set. The 82576 stores the SMBus address that was assigned in the SMBus-ARP process in its EEPROM, so after the next power-up, it returns to its assigned SMBus address.

Figure 3-4 shows the SMBus-ARP behavior of the 82576.

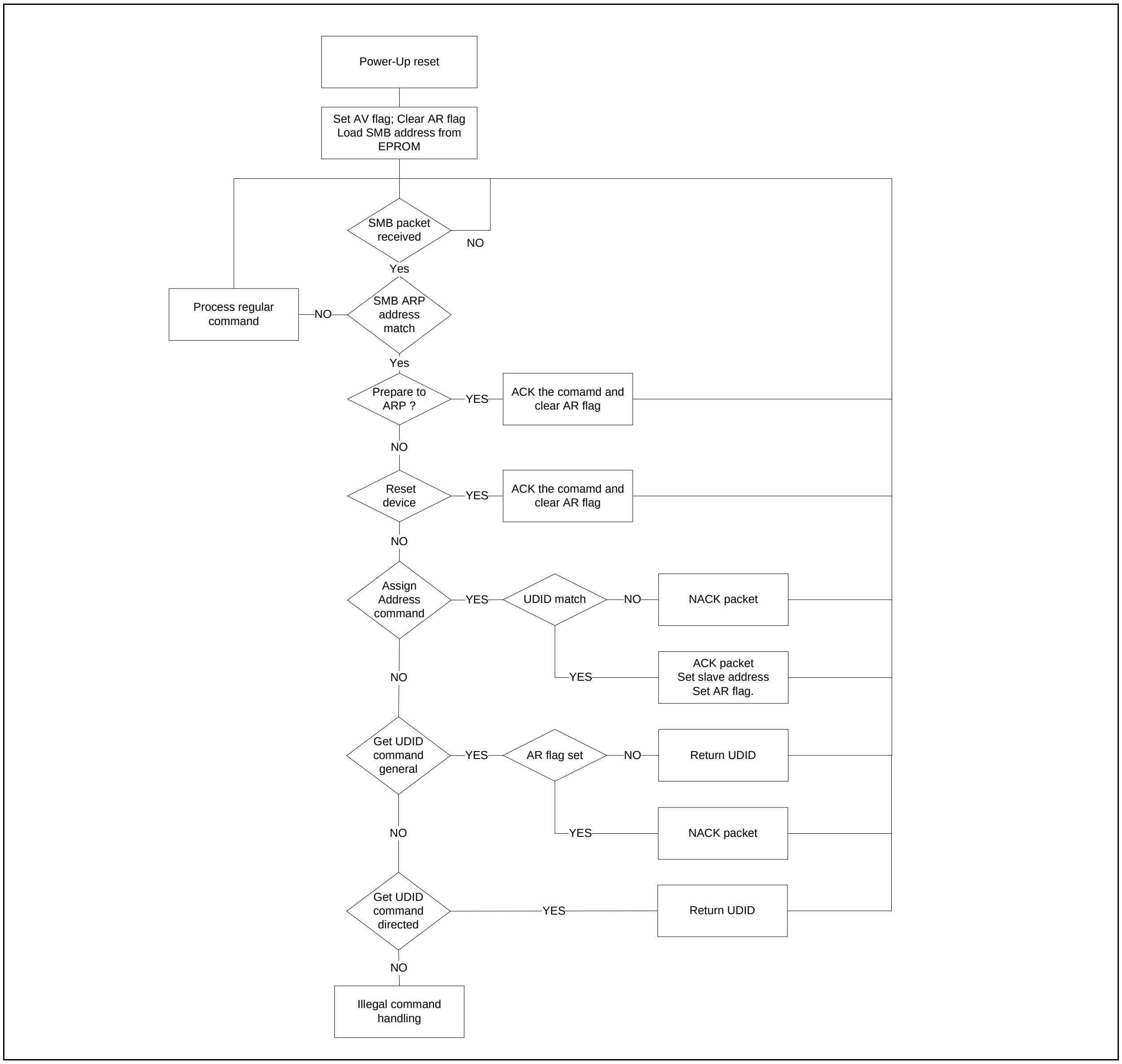


Figure 3-4. SMBus ARP Flow

3.2.1.1.8.3 SMBus ARP UDID Content

The Unique Device Identifier (UDID) provides a mechanism to isolate each device for the purpose of address assignment. Each device has a unique identifier. The 128-bit number is comprised of the following fields:

Table 3-17. Unique Device Identifier (UDID)

1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	4 Bytes

Table 3-17. Unique Device Identifier (UDID)

Device Capabilities	Version / Revision	Vendor ID	Device ID	Interface	Sub-system Vendor ID	Sub- system Device ID	Vendor Specific ID
See below	See below	0x8086	0x10AA	0x0004	0x0000	0x0000	See below
MSB							LSB

Where:

- Vendor ID — The device manufacturer's ID as assigned by the SBS Implementers' Forum or the PCI SIG — Constant value: 0x8086.
- Device ID — The device ID as assigned by the device manufacturer (identified by the *Vendor ID* field) - Constant value: 0x10AA.
- Interface — Identifies the protocol layer interfaces supported over the SMBus connection by the device - In this case, SMBus Version 2.0 - Constant value: 0x0004.
- Sub-system Fields — These fields are not supported and return zeros.

Device Capabilities: Dynamic and Persistent Address, PEC Support bit:

Table 3-18. Dynamic and Persistent Address, PEC Support bit

7	6	5	4	3	2	1	0
Address Type		Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	PEC Supported
0b	1b	0b	0b	0b	0b	0b	0b
MSB							LSB

Version/Revision: UDID Version 1, Silicon Revision:

Table 3-19. Version/Revision: UDID Version 1, Silicon Revision

7	6	5	4	3	2	1	0
Reserved (0)	Reserved (0)	UDID Version			Silicon Revision ID		
0b	0b	001b			See below		
MSB							LSB

Silicon Revision ID:

Table 3-20. Silicon Revision ID

Silicon version	Revision ID
A1	001b

Vendor Specific ID - Four LSB bytes of the 82576's Ethernet MAC address. The 82576's Ethernet address is taken from words 0b-2b in the EEPROM. Note that in the 82576 there are two MAC addresses (one for each port). Bit 0 of the port 1 MAC address has the inverted value of bit 0 from the EEPROM.

Table 3-21. Vendor Specific ID

1 Byte	1 Byte	1 Byte	1 Byte
MAC Address, byte 3	MAC Address, byte 2	MAC Address, byte 1	MAC Address, byte 0
MSB			LSB

3.2.1.1.9 LAN Fail-Over Through SMBus

In fail-over mode, the 82576 determines which ports are used for transmit and receive (according to the configuration). LAN fail-over is tied to the SMBus addressing mode. When the SMBus is dual-address mode, the 82576 does not activate its fail-over mechanism (ignores the fail-over register) and operates using individual LAN ports. When the SMBus is in single-address mode or in pass-through mode, the 82576 operates in fail-over mode as described in Section 10.2.2.2.

3.2.2 NC-SI

The NC-SI interface in the 82576 is a connection to an external BMC defined by the DMTF NC-SI protocol. It operates as a single interface with an external BMC, where all traffic between the 82576 and the BMC flows through the interface.

3.2.2.1 Electrical Characteristics

The 82576 complies with the electrical characteristics defined in the NC-SI specification. However, the 82576 pads are not 5V tolerant and require that signals conform to 3.3V signaling.

The 82576 NC-SI behavior is configured by the 82576 on power-up:

- The 82576 provides an NC-SI clock output if enabled by the *NC-SI Clock Direction* EEPROM bit. The default value is to use an external clock source as defined in the NC-SI specification.
- The output driver strength for the NC-SI_CLK_OUT pad is configured by the EEPROM *NC-SI Clock Pad Drive Strength* bit (default = 0b).
- The output driver strength for the NC-SI output signals (NC-SI_DV & NC-SI_RX) is configured by the EEPROM *NC-SI Data Pad Drive Strength* bit (default = 0b).
- The *Multi-Drop NC-SI EEPROM* bit defines the NC-SI topology (point-to-point or multi-drop; the default is point-to-point).

The 82576 can provide an NC-SI clock output as previously mentioned. The NC-SI clock input (NC-SI_CLK_IN) serves as an NC-SI input clock in either case. That is, if the 82576 provides an NC-SI output clock, the platform is required to route it back through the NC-SI clock input with the correct latency. See the Electrical chapter for more details.

The 82576 dynamically drives its NC-SI output signals (NC-SI_DV and NC-SI_RX) as required by the sideband protocol:

- On power-up, the 82576 floats the NC-SI outputs
- If the 82576 operates in point-to-point mode, then the 82576 starts driving the NC-SI outputs at some time following power-up
- If the 82576 operates in a multi-drop mode, the 82576 drives the NC-SI outputs as configured by the BMC.

3.2.2.2 NC-SI Transactions

The NC-SI link supports both pass-through traffic between the BMC and the 82576 LAN functions, as well as configuration traffic between the BMC and the 82576 internal units as defined in the NC-SI protocol. See Section 10.2.1 and Section 10.3.1 in this manual.

3.3 Flash / EEPROM

3.3.1 EEPROM Interface

3.3.1.1 General Overview

The 82576 uses an EEPROM device for storing product configuration information. The EEPROM is divided into three general regions:

- Hardware accessed - Loaded by the 82576 after power-up, PCI reset de-assertion, D3 ->D0 transition, or a software-commanded EEPROM read (CTRL_EXT.EE_RST).
- Manageability firmware accessed - Loaded by the 82576 in pass-through mode after power-up or firmware reset.
- Software accessed - Used only by software. The meaning of these registers, as listed here, is a convention for software only and is ignored by the 82576.

Table 3-22 lists the structure of the EEPROM image in the 82576.

Table 3-22. EEPROM Structure

Address	Content
0x0 – 0x9	MAC address and software area
0xA – 0x2F	Hardware area (+ pointer to analog configuration)
0x30 – 0x3F	PXE area
0x40 – 0x4F	Reserved
0x50 – 0x5A	FW pointers
...	
	Firmware structures
...	
	VPD area
...	
	Analog configuration (PCIe/PHY/PLL/SerDes structures)

The EEPROM mapping is described in Section 6..

3.3.1.2 EEPROM Device

The EEPROM interface supports an SPI interface and expects the EEPROM to be capable of 2 MHz operation.

The 82576 is compatible with many sizes of 4-wire serial EEPROM devices. If pass-through mode functionality is desired, up to a 256 Kbytes serial SPI compatible EEPROM can be used. If no manageability mode is desired, a 4096-bit serial SPI compatible EEPROM can be used. All EEPROM's are accessed in 16-bit words although the EEPROM is designed to also accept 8-bit data accesses.

The 82576 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*) field appropriately. Software can use this size to determine how to access the EEPROM. The exact size of the EEPROM is determined within one of the EEPROM words.

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

3.3.1.3 Software Accesses

The 82576 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.

In addition, the VPD area of the EEPROM can be accessed via the VPD capability structure of the PCIe.

Software can use the EEPROM Read (EERD) register to cause the 82576 to read a word from the EEPROM that the software can then use. To do this, software writes the address to read to the *Read Address* (EERD.ADDR) field simultaneously writes a 1b to the *Start Read* bit (EERD.START). The 82576 reads the word from the EEPROM, sets the *Read Done* bit (EERD.DONE), and puts the data in the *Read Data* field (EERD.DATA). Software can poll the EEPROM Read register until it sees the *Read Done* bit set and then uses the data from the *Read Data* field. Any words read this way are not written to the 82576's internal registers.

Software can also directly access the EEPROM's 4-wire interface through the EEPROM/Flash Control (EEC) register. 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 Reques*t bit (EEC.EE_REQ).
2. Read the *EEPROM Grant* bit (EEC.EE_GNT) until it becomes 1b. It remains 0b as long as the hardware is accessing the EEPROM.
3. Write or read the EEPROM using the direct access to the 4-wire interface as defined in the EEPROM/ Flash Control and Data (EEC) register. 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).

Finally, software can cause the 82576 to re-read the hardware accessed fields of the EEPROM (setting the 82576's internal registers appropriately) by writing a 1b to the *EEPROM Reset* bit of the Extended Device Control register (CTRL_EXT.EE_RST).

Note: If the EEPROM does not contain a valid signature (see Section 3.3.1.4), the 82576 assumes 16-bit addressing. In order to access an EEPROM that requires 8-bit addressing, software must use the direct access mode.

3.3.1.4 Signature Field

The 82576 determines if an EEPROM is present by attempting to read it. The 82576 first reads the EEPROM Sizing and Protected Fields word at address 0x12. It 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 then programs its internal registers based on the values read. Otherwise, it ignores the values it reads from that location and does not read any other words as part of the auto-read process. However, the EEPROM is still accessible to software.

3.3.1.5 Protected EEPROM Space

The 82576 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 subsystem. This area is located at the end of the EEPROM memory. It’s size is defined by the HEPSize field in EEPROM word 0x12. Note that the current the 82576 firmware does not use this mechanism.

A mechanism to protect part of the EEPROM from host writes is also provided. This mechanism is controlled by word 0x2D and 0x2C that controls the start and the end of the read-only area.

3.3.1.5.1 Initial EEPROM Programming

In most applications, initial EEPROM programming is done directly on the EEPROM pins. Nevertheless, it is desired to enable existing software utilities (accessing the EEPROM via the host interface) to initially program the entire EEPROM without breaking the protection mechanism. Following a power-up sequence, the 82576 reads the hardware initialization words in the EEPROM. If the signature in word 0x12 does not equal 01b, the EEPROM is assumed as non-programmed. There are two effects of a non-valid signature:

- The 82576 does not read any further EEPROM data and sets the relevant registers to default.
- The 82576 enables access to any location in the EEPROM via the EEPROM CSR registers.

3.3.1.5.2 Activating the Protection Mechanism

Following initialization, the 82576 reads the EEPROM and turns on the protection mechanism if word 0x12 contains a valid signature (equals 01b) and word 0x12, bit 4 is set (enable protection). Once the protection mechanism is turned on, words 0x12, 0x2C and 0x2D become write-protected, the area that is defined by word 0x12 becomes hidden (such as read/write protected) and the area defined by words 0x2C and 0x2D become write protected.

- No matter what is designated as the read only protected area, words 0x30:0x3F (used by PXE driver) are writeable, unless it is defined as hidden.

3.3.1.5.3 Non Permitted Accessing to Protected Areas in the EEPROM

This paragraph refers to EEPROM accesses via the EEC (bit banging) or EERD (parallel read access) registers. Following a write access to the protected areas in the EEPROM, hardware responds properly on the PCIe interface but does not initiate any access to the EEPROM. Following a read access to the hidden area in the EEPROM (as defined by word 0x12), hardware does not access the EEPROM and returns meaningless data to the host.

Note: Using bit banging, the SPI EEPROM can be accessed in a burst mode. For example, providing op-code, address, and then read or write data for multiple bytes. Hardware inhibits any attempt to access the protected EEPROM locations even in burst accesses.

Software should not access the EEPROM in a burst-write mode starting in a non-protected area and continue to a protected one. In such a case it is not guaranteed that the write access to any area ever takes place.

3.3.1.6 EEPROM-Less Support

The 82576 supports EEPROM-less operation with the following limitations:

- Non-manageability mode only.
- No support for legacy Wake on LAN (magic packets).
- No support for Flash (such as no PXE code on a LOM/NIC).
- No support for serial ID PCIe capability.
- All initialization values usually taken from the EEPROM should be done by the host driver.

3.3.1.6.1 Access to the EEPROM Controlled Feature

The z register enables access to registers that are not accessible via regular CSR access (such as PCIe configuration read-only registers) by emulating the auto-read process. EEARBC contains three strobe fields that emulate the internal strobes of the internal auto-read process. This register is common to both functions and should be accessed only after the coordination with the other port.

Table 3-23 lists the strobe to be used when emulating a read of a specific word of the EEPROM auto-read feature.

Table 3-23. Strobes for EEARBC Auto-Read Emulation

EEPROM Word Emulated (In Hex)	Content	Strobe for Port 0	Strobe for Port 1
0:2	MAC address	VALID_CORE0	VALID_CORE1
0A/0F	Init control 1/2	VALID_CORE0	VALID_CORE1
0B/0C[1]	Sub-system device and vendor	VALID_COMMON	VALID_COMMON
1E/1D[2]	Dummy device ID, Rev ID	VALID_COMMON	VALID_COMMON
21	Function control	VALID_COMMON	VALID_COMMON
0D[2]	Device ID port 0	VALID_COMMON	N/A
11[2]	Device ID port 1	N/A	VALID_COMMON
10	SDP control	N/A	VALID_CORE1
20	SDP control	VALID_CORE0	N/A
14	Init control 3	N/A	VALID_CORE1
24	Init control 3	VALID_CORE0	N/A
15/16/18/19/1A/1B/ 22/25/26	PCIe and NC-SI configuration	VALID_COMMON	VALID_COMMON
1C/1F	LED control port 0	VALID_CORE0	N/A
2A/2B[3]	LED control port 1	N/A	VALID_CORE1
2E[3]	Watchdog configuration	VALID_CORE0	VALID_CORE1
2F	VPD area	N/A	N/A

[1] If word 0xA was accessed before the subsystem or subvendor ID are set, care must be taken that the load Subsystem IDs bit in word 0xA is set.
[2] If word 0xA was accessed before one of the device IDs is set, care must be taken that the load Device IDs bit in word 0xA is set.
[3] Part of the parameters that can be configured through the EEARBC register can be directly set through regular registers and thus usage of this mechanism is not needed for them. Specifically, words 0x2A, 0x2B and 0x2E controls only parameters that can be set through regular registers.

3.3.2 Shared EEPROM

The 82576 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, wake-up capability, etc. Certain EEPROM words are used to specify hardware parameters that are LAN device-independent (such as those that affect circuit behavior). Other EEPROM words are associated with a specific LAN device. Both LAN devices access the EEPROM to obtain their respective configuration settings.

3.3.2.1 EEPROM Deadlock Avoidance

The EEPROM is a shared resource between the following clients:

- Hardware auto-read.
- Port 0 LAN driver accesses.
- Port 1 LAN driver accesses.

- Firmware accesses.

All clients can access the EEPROM using parallel access, where hardware implements the actual access to the EEPROM. Hardware can schedule these accesses so that all clients get served without starvation.

However, software and hardware clients can access the EEPROM using bit banging. In this case, there is a request/grant mechanism that locks the EEPROM to the exclusive usage of one client. If this client is stuck (without releasing the lock), the other clients are not able to access the EEPROM. In order to avoid this, the 82576 implements a timeout mechanism, which releases the grant from a client that didn't toggle the EEPROM bit-bang interface for more than two seconds.

Note: If an agent that was granted access to the EEPROM for bit-bang access didn't toggle the bit bang interface for 500 ms, it should check if it still owns the interface before continuing the bit-banging.

3.3.2.2 EEPROM Map Shared Words

The EEPROM map in Section 6.1 identifies those words configuring either LAN devices or the entire Intel® 82576 GbE Controller component as “both”. Those words configuring a specific LAN device parameter are identified by their LAN number.

The following EEPROM words warrant additional notes specifically related to dual-LAN support:

Table 3-24. Notes on EEPROM Words

Ethernet Address (IA) (shared between LANs)	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 0x010000000000}.
Initialization Control 1, Initialization Control 2 (shared between LANs)	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, wake-up settings, PCIe bus advertised capabilities, etc.
Initialization Control 3 (unique to each LAN)	This EEPROM word configures default values associated with each LAN device’s hardware connections, including which link mode (internal PHY, SGMII, 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.

3.3.3 Vital Product Data (VPD) Support

The EEPROM image might contain an area for VPD. This area is managed by the OEM vendor and doesn’t influence the behavior of hardware. Word 0x2F of the EEPROM image contains a pointer to the VPD area in the EEPROM. A value of 0xFFFF means VPD is not supported and the VPD capability doesn’t appear in the configuration space.

The VPD area should be aligned to a Dword boundary in the EEPROM and should start in the first 1Kbyte of the EEPROM.

The maximum area size is 256 bytes but can be smaller. The VPD block is built from a list of resources. A resource can be either large or small. The structure of these resources are listed in the following tables.

Table 3-25. Small Resource Structure

Offset	0	1 - n
Content	Tag = 0xxx, xyyyb (Type = Small(0), Item Name = xxxx, length = yyy bytes)	Data

Table 3-26. Large Resource Structure

Offset	0	1 - 2	3 - n
Content	Tag = 1xxx, xxxxb (Type = Large(1), Item Name = xxxxxxxx)	Length	Data

The 82576 parses the VPD structure during the auto-load process (power up and PCIe reset or warm reset) in order to detect the read-only and read/write area boundaries. The 82576 assumes the following VPD structure:

Table 3-27. VPD Structure

Tag	Structure Type	Length (Bytes)	Data	Resource Description
0x82	Large	Length of identifier string	Identifier	Identifier string.
0x90	Large	Length of RO area	RO data	VPD-R list containing one or more VPD keywords This part is optional and might not appear.
0x91	Large	Length of R/W area	RW data	VPD-W list containing one or more VPD keywords. This part is optional and might not appear.
0x78	Small	N/A	N/A	End tag.

Note: The VPD-R and VPD-W structures can be in any order.

If the 82576 doesn't detect a value of 0x82 in the first byte of the VPD area, or the structure doesn't follow the description listed in Table 3-27, it assumes the area is not programmed and the entire 256 bytes area is read only. If a VPD-W tag is found after the VPD-R tag, the area defined by it's size is writable via the VPD structure. Refer to the PCI 3.0 specification (Appendix I) for details of the different tags.

In any case, the VPD area is accessible for read and write via the regular EEPROM mechanisms pending the EEEPROM protection capabilities enabled. For example, if VPD is in the protected area, the VPD area is not accessible to the software device driver (parallel or serial), but accessible through the VPD mechanism. If the VPD area is not in the protected area, then the software device driver can access all of it for read and write.

The VPD area can be accessed through the PCIe configuration space VPD capability structure described in Section 9.5.4. Write accesses to a read-only area or any access outside of the VPD area via this structure are ignored.

Note: Write access to Dwords, which are only partially in the read/write area, are ignored. It is responsibility of VPD software to make the right alignment to enable a write to the entire area.

3.3.4 Flash Interface

3.3.4.1 Flash Interface Operation

The 82576 provides two different methods for software access to the Flash.

Using the legacy Flash transactions, the Flash is read from or written to each time the host CPU performs a read or a write operation to a memory location that is within the Flash address mapping or after a re-boot via accesses 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. Refer to the specific Flash data sheet for more details on reading from or writing to Flash. Accesses to the Flash are based on a direct decode of CPU accesses to a memory window defined in either:

1. The 82576's Flash Base Address register (PCIe Control register at offset 0x14 or 0x18).
2. A certain address range of the IOADDR register defined by the IO Base Address register (PCIe Control register at offset 0x18 or 0x20).
3. The Expansion ROM Base Address register (PCIe Control register at offset 0x30).

The 82576 controls accesses to the Flash when it decodes a valid access.

Note: Flash read accesses must always be assembled by the 82576 each time the access is greater than a byte-wide access.

The 82576 byte reads or writes to the Flash take on the order of 2 µs. The 82576 continues to issue retry accesses during this time.

The 82576 supports only byte writes to the Flash.

Another way for software to access the Flash is directly using the Flash's 4-wire interface through the Flash Access (FLA) register. It can use this for reads, writes, or other Flash operations (accessing the Flash status register, erase, etc.).

To directly access the Flash, software should follow 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.
3. Write or read the Flash using the direct access to the 4-wire interface as defined in the FLA register. 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).

3.3.4.2 Flash Write Control

The Flash is write controlled by the *FWE* bits in the EEPROM/FLASH Control and Data (EEC) register. Note that attempts to write to the Flash device when writes are disabled (FWE=10b) should not be attempted. Behavior after such an operation is undefined and can result in component and/or system hangs.

After sending one byte write to the Flash, software checks if it can send the next byte to write (check if the write process in the Flash had finished) by reading the FLA register, If bit (FLA.FL_BUSY) in this register is set, the current write did not finish. If bit (FLA.FL_BUSY) is clear then software can continue and write the next byte to the Flash.

3.3.4.3 Flash Erase Control

When software needs to erase the Flash, it should set bit *FLA.FL_ER* in the FLA register to 1b (Flash erase) and then set bits *EEC.FWE* in the EEPROM/Flash Control register to 0b.

Hardware gets this command and sends the Erase command to the Flash. The erase process finishes by itself. Software should wait for the end of the erase process before any further access to the Flash. This can be checked by using the Flash write control mechanism previously described.

The op-code used for erase operation is defined in the FLASHOP register.

Note: Sector erase by software is not supported. In order to delete a sector, the serial (bit bang) interface should be used.

3.3.5 Shared FLASH

The 82576 provides an interface to an external serial Flash/ROM memory device, as described in Section 2.1.2. This Flash/ROM device can be mapped into memory and/or I/O address space for each LAN device through the use of Base Address Registers (BARs). Bit 13 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 write ability.

3.3.5.1 Flash Access Contention

The 82576 implements internal arbitration between Flash accesses initiated through the LAN 0 device and those initiated through the LAN 1 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: The 82576 does not synchronize between the two entities accessing the Flash. Contentions caused by one entity reading and the other modifying the same location is possible.

To avoid this contention, accesses from both LAN devices should 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 by the nature of the software sequence.

3.3.5.2 Flash Deadlock Avoidance

The Flash is a shared resource between the following clients:

- Port 0 LAN driver accesses.
- Port 1 LAN driver accesses.
- BIOS parallel access via expansion ROM mechanism.
- Firmware accesses.

All clients can access the flash using parallel access, where hardware implements the actual access to the Flash. Hardware can schedule these accesses so that all the clients get served without starvation.

However, the driver and firmware clients can access the serial Flash using bit banging. In this case, there is a request/grant mechanism that locks the serial Flash to the exclusive usage of one client. If this client is stuck without releasing the lock, the other clients are unable to access the Flash. In order to avoid this, the 82576 implements a time-out mechanism that releases the grant from a client that doesn't toggle the Flash bit-bang interface for more than two seconds.

Note: If an agent that was granted access to the Flash for bit-bang access doesn't toggle the bit-bang interface for 500 ms, it should check that it still owns the interface before continuing the bit banging.

This mode is enabled by bit five in word 0xA of the EEPROM.

3.4 Configurable I/O Pins

3.4.1 General-Purpose I/O (Software-Definable Pins)

The 82576 has four software-defined pins (SDP pins) per port that can be used for miscellaneous hardware or software-controllable purposes. These pins and their function are bound to a specific LAN device. For example, eight SDP pins cannot be associated with a single LAN device. These pins can each be individually configurable to act as either input or output pins. The default direction of each of the four pins is configurable via the EEPROM as well as the default value of any pins configured as outputs. To avoid signal contention, all four pins are set as input pins until after the EEPROM configuration has been loaded.

In addition to all four pins being individually configurable as inputs or outputs, they can be configured for use as General-Purpose Interrupt (GPI) inputs. To act as GPI pins, the desired pins must be configured as inputs. A separate GPI interrupt-detection enable is then used to enable rising-edge detection of the input pin (rising-edge detection occurs by comparing values sampled at the internal clock rate as opposed to an edge-detection circuit). When detected, a corresponding GPI interrupt is indicated in the Interrupt Cause register.

The use, direction, and values of SDP pins are controlled and accessed using fields in the Device Control (CTRL) register and Extended Device Control (CTRL_EXT) register.

The SDPs can be used for special purpose mechanism such as watch dog indication (see Section 3.4.2 for details) or IEEE 1588 support (see Section 7.10.4 for details).

3.4.2 Software Watchdog

In some situations it might be useful to give an indication to manageability firmware or to external devices that the 82576 hardware or the software device driver is not functional. For example, in a pass-through NIC, the 82576 might be bypassed if it is not functional. In order to provide this functionality, a watchdog mechanism is used. This mechanism can be enabled by default, according to EEPROM configuration. Once the host driver is up and it determines that hardware is functional, it might reset the watchdog timer to indicate that the 82576 is functional. The software device driver then should re-arm the timer periodically. If the timer is not re-armed after pre-programmed timeout (such as the timer runs out before it is reset), an interrupt is given to firmware and a pre-programmed SDP (either SDP0[0] or SDP1[0]) is raised. Note that the SDP indication is shared between the ports.

The register controlling this feature is WDSETUP. This register enables the setting of a time-out period and the activation of this mode. Both get their default from the EEPROM.

The re-arming of the timer is done by setting the *WDSWSTS.Dev_functional* bit.

If software needs to trigger the watchdog immediately because it suspects hardware is stuck, it can set the *WDSWSTS.Force_WD* bit. It can also give firmware an indication of the watchdog reason for using the *WDSWSTS.stuck_reason* field.

The SDP on which the watchdog indication is indicated, is set using the *CTRL.SDP0_WDE bit.* In this mode the *CTRL.SDP0_IODIR* should be set to output. The *CTRL.SDP0_DATA* bit indicates the polarity of the indication. Setting this bit in one of the cores causes the watchdog indications of both ports to be indicated on this SDP.

3.4.2.1 Watchdog rearm

After a watchdog indication was received, in order to rearm the mechanism the following flow should be used:

1. Clear WD_enable bit in the WDSTP register.
2. Clear SDP0_WDE bit in CTRL register.
3. Set SDP0_WDE bit in CTRL register.
4. Set WD_enable in the WDSTP register.

3.4.3 LEDs

The 82576 provides four LEDs per port that can be used to indicate different statuses of the traffic. The default setup of the LEDs is done via EEPROM words 0x1C, 0x1F for port 0 and words 0x2A, 0x2B for port 1. This setup is reflected in the LEDCTL register of each port. Each software device driver can change its setup individually. For each of the LEDs the following parameters can be defined:

- Mode: Defines which information is reflected by this LED. The encoding is described in the LEDCTL register.
- Polarity: Defines the polarity of the LED.
- Blink mode: Determines whether or not the LED should blink or be stable.

In addition, the blink rate of all LEDs can be defined. The possible rates are 200 ms or 83 ms for each phase. There is one rate for all the LEDs of a port.

3.5 Network Interfaces

3.5.1 Overview

The 82576 MAC provides a complete CSMA/CD function supporting IEEE 802.3 (10 Mb/s), 802.3u (100 Mb/s), 802.3z and 802.3ab (1000 Mb/s) implementations. The 82576 performs all of the functions required for transmission, reception, and collision handling called out in the standards.

Each 82576 MAC can be configured to use a different media interface. The 82576 supports the following potential configurations:

- Internal copper PHY.
- External SerDes device such as an optical SerDes (SFP or on board) or backplane connections.
- External SGMII device. This mode is used for SFP connections or external SGMII PHYs.

Selection between the various configurations is programmable via each MAC's Extended Device Control register (*CTRL_EXT.LINK_MODE* bits) and default is set via EEPROM settings. Table 3-28 lists the encoding on the *LINK_MODE* field for each of the modes.

Table 3-28. Link Mode Encoding

Link Mode	82576 Mode
00b	Internal PHY
01b	Reserved
10b	SGMII
11b	SerDes

The GMII/MII interface used to communicate between the MAC and the internal PHY or the SGMII PCS 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 SerDes function can be used to implement a fiber-optics-based solution or backplane connection without requiring an external TBI mode transceiver/SerDes.

The SGMII interface can be used to connect to SFP modules. As such, this SGMII interface has the following limitations:

- No Tx clock
- AC coupling only

The internal copper 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 the ability to auto-negotiate to lower-speed (and less power-hungry) 10/100 Mb/s operation when the system is in low power-states.

3.5.2 MAC Functionality

3.5.2.1 Internal GMII/MII Interface

The 82576's MAC and PHY/PCS 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 the 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.

3.5.2.2 MDIO/MDC

The 82576 implements an IEEE 802.3 MII Management Interface (also known as the Management Data Input/Output or MDIO Interface) between the MAC and the PHY. This interface provides the MAC and software the ability to monitor and control the state of the PHY. The MDIO interface defines a physical connection, a special protocol that runs across the connection, and an internal set of addressable registers. The internal or external 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 is 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 can use MDIO accesses to read or write registers in internal PHY mode by accessing the 82576's MDIC register (see Section 8.2.4).

When working in SGMII/SerDes mode, the external PHY (if it exists) can be accessed either through MDC/MDIO as previously described, or via a two wire interface bus using the I2CCMD register (see Section 8.18.8). The two wire interface bus or the MDC/MDIO bus are connected via the same pins, and thus are mutually exclusive. In order to be able to control an external device, either by SFP or MDC/MDIO, the I^{2}C *SFP Enable* bit in Initialization Control 3 EEPROM word should be set.

As the MDC/MDIO command can be targeted either to the internal PHY or to an external bus, the *MDIC.destination* bit is used to define the target of the transaction.

Note: Each port has its own MDC/MDIO or two wire interface bus and there is no sharing between the ports of the control port. In order to control both ports' PHYs, via the same control bus, accesses to both PHYs should be done via the same port with different device addresses.

3.5.2.2.1 MDIC register usage

For an MDI read cycle, the sequence of events is as follows:

1. The processor 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).
2. 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.
3. The PHY returns the following sequence on the MDIO signal:

   <0><DATA><IDLE>.
4. The MAC discards the leading bit and places the following 16 data bits in the MII register.
5. The the 82576 asserts an interrupt indicating MDI “Done” if the *Interrupt Enable* bit was set.
6. The the 82576 sets the *Ready* bit in the MII register indicating the Read is complete.
7. The processor might read the data from the MII register and issue a new MDI command.

For a MDI write cycle, the sequence of events is as follows:

1. Ready = 0b.
2. Interrupt Enable set to 1b or 0b.
3. Opcode = 01b (write).
4. PHYADD = PHY address from the MDI register.
5. REGADD = Register address of the specific register to be accessed (0 through 31).
6. Data = Specific data for desired control of the PHY.
7. The MAC applies the following sequence on the MDIO signal to the PHY:

   ```
   <PREAMBLE><01><01><PHYADD><REGADD><10><DATA><IDLE>
   ```
8. The the 82576 asserts an interrupt indicating MDI “Done” if the *Interrupt Enable* bit was set.
9. The the 82576 sets the *Ready* bit in the MII register to indicate that the write operation completed.
10. The CPU might issue a new MDI command.

Note: An MDI read or write might take as long as 64 µs from the processor 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 0xFFFF to the data field to indicate an error condition, and sets the *Ready* bit.

Note: After a PHY reset, access through the MDIC register should not be attempted for 300 µsec.

3.5.2.3 Duplex Operation with Copper PHY

The 82576 supports half-duplex and full-duplex 10/100 Mb/s MII mode either through the internal copper PHY or SGMII interface. However, only full-duplex mode is supported when SerDes mode is used or in any 1000 Mb/s connection.

Configuration of the duplex operation of the 82576 can either be forced or determined via the auto-negotiation process. See Section 3.5.4.3 for details on link configuration setup and resolution.

3.5.2.3.1 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 82576 and the specific capabilities of the link partner used in the application. During full-duplex operation, the 82576 can transmit and receive packets simultaneously across the link interface.

In full-duplex, transmission and reception are delineated independently by the GMII/MII control signals. Transmission starts TX_EN is asserted, which indicates there is valid data on the TX_DATA bus driven from the MAC to the PHY/PCS. Reception is signaled by the PHY/PCS by the asserting the RX_DV signal, which indicates valid receive data on the RX_DATA lines to the MAC.

3.5.2.3.2 Half Duplex

In half-duplex operation, the MAC attempts to avoid contention with other traffic on the link by monitoring the CRS signal provided by the PHY and deferring to passing traffic. When the CRS signal is de-asserted or after a sufficient Inter-Packet Gap (IPG) has elapsed after a transmission, frame transmission begins. The MAC signals the PHY/PCS with TX_EN at the start of transmission.

In the case of a collision, the PHY/SGMII detects the collision and asserts the COL signal to the MAC. Frame transmission stops within four link clock times and then the 82576 sends a JAM sequence onto the link. After the end of a collided transmission, the 82576 backs off and attempts to re-transmit per the standard CSMA/CD method.

Note: The re-transmissions are done from the data stored internally in the 82576 MAC transmit packet buffer (no re-access to the data in host memory is performed).

The MAC behavior is different if a regular collision or a late collision is detected. If a regular collision is detected, the MAC always tries to re-transmit until the number of excessive collisions is reached. In case of late collision, the MAC retransmission is configurable. In addition, statistics are gathered on late collisions.

In the case of a successful transmission, the 82576 is ready to transmit any other frame(s) queued in the MAC's transmit FIFO, after the minimum inter-frame spacing (IFS) of the link has elapsed.

During transmit, the PHY is expected to signal a carrier-sense (assert the CRS signal) back to the MAC before one slot time has elapsed. The transmission completes successfully even if the PHY fails to indicate CRS within the slot time window. If this situation occurs, the PHY can either be configured incorrectly or be in a link down situation. Such an event is counted in the Transmit without CRS statistic register (see Section 8.19.11).

3.5.3 SerDes, SGMII Support

The 82576 can be configured to follow either SGMII, SerDes standards. When in SGMII mode, the 82576 can be configured to operate in 1 Gb/s, 100 Mb/s or 10 Mb/s speeds. When in the 10/100 Mb/s speed, they can be configured to half-duplex mode of operation. When configured for SerDes operation, the port supports only 1 Gb/s, full-duplex operation. Since the serial interfaces are defined as differential signals, internally the hardware has analog and digital blocks. Following is the initialization/configuration sequence for the analog and digital blocks.

3.5.3.1 SerDes Analog Block

The analog block may require some changes to it's configuration registers in order to work properly. There is no special requirement for designers to do these changes as the hardware internally updates the configuration using a default sequence or a sequence loaded from the EEPROM. There is a provision for EEPROM-less systems, where software can generate the same changes that the hardware generates by writing the initialization sequence through the SCCTL register.

3.5.3.2 SerDes/SGMII PCS Block

The link setup for SerDes and SGMII are described in sections 3.5.4.1 and 3.5.4.2, respectively.

3.5.3.3 GbE Physical Coding Sub-Layer (PCS)

The 82576 integrates the 802.3z PCS function on-chip. The on-chip PCS circuitry is used when the link interface is configured for SerDes or SGMII operation and is bypassed for internal PHY mode.

The packet encapsulation is based on the Fiber Channel (FC0/FC1) physical layer and uses the same coding scheme to maintain transition density and DC balance. The physical layer device is the SerDes and is used for 1000BASE-SX, -L-, or -CX configurations.

3.5.3.3.1 8B10B Encoding/Decoding

The GbE PCS circuitry uses the same transmission-coding scheme used in the fiber channel physical layer specification. The 8B10B-coding scheme was chosen by the 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% overhead for this transmission code, which accounts for the data-signaling rate of 1250 Mb/s with 1000 Mb/s of actual data.

3.5.3.3.2 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 GbE packets. The following table contains a brief description of defined ordered sets and included for informational purposes only. See clause 36 of the IEEE 802.3z specification for more details.

Table 3-29. Brief Description of Defined Ordered Sets

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 and negotiate link operation information between link partners. Last 2 code groups contain configuration base and next page registers.
/C1/	Configuration 1	4	See /C/. Differs from /C2/ in 2nd code group for maintaining proper signaling disparity[1].
/C2/	Configuration 2	4	See /C/. Differs from /C1/ in 2nd code group for maintaining proper signaling disparity[1].
/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 enable the SerDes to maintain clock and symbol synchronization between link partners.
/I1/	IDLE 1	2	See /I/. Differs from /I2/ in 2nd code group for maintaining proper signaling disparity[1].

Table 3-29. Brief Description of Defined Ordered Sets (Continued)

/I2/	IDLE 2	2	See /I/. Differs from /I1/ in 2nd code group for maintaining proper signaling disparity[1].
/R/	Carrier_Extend	1	This ordered set is used to indicate carrier extension to the receiving PCS. It is also used as part of the end_of_packet encapsulation delimiter as well as IPG for packets in a burst of packets.
/S/	Start_of_Packet	1	The SPD (start_of_packet delimiter) ordered set is used to indicate the starting boundary of a packet transmission. This symbol replaces the last byte of the preamble received from the MAC layer.
/T/	End_of_Packet	1	The EPD (end_of_packet delimiter) is comprised of three ordered sets. The /T/ symbol is always the first of these and indicates the ending boundary of a packet.
/V/	Error_Propagation	1	The /V/ ordered set is used by the PCS to indicate error propagation between stations. This is normally intended to be used by repeaters to indicate collisions.

[1] The concept of running disparity is defined in the standard. In summary, this refers to the 1-0 and 0-1 transitions within 8B10B code groups.

3.5.4 Auto-Negotiation and Link Setup Features

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 82576 provides the complete 802.3z PCS function. For internal PHY mode, the PCS and auto-negotiation functions are maintained within the PHY. For SGMII mode, the 82576 supports the SGMII link auto-negotiation process, whereas the link auto-negotiation is done by the external PHY.

Configuring the link can be accomplished by several methods ranging from software 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 82576 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 (internal PHY, external SerDes or SGMII) being used.

When operating in a SerDes mode, the PCS layer 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 82576.

When operating in internal PHY mode, 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 internal PHY to the MAC, or by MAC auto-detection functions.

When operating in SGMII mode, the PCS layer performs SGMII auto-negotiation per the SGMII specification. The external PHY is responsible for the Ethernet auto-negotiation process.

3.5.4.1 SerDes Link Configuration

When using SerDes link mode, link mode configuration can be performed using the PCS function in the 82576. The hardware supports both hardware and software auto-negotiation methods for determining the link configuration, as well as allowing for a manual configuration to force the link. Hardware auto-negotiation is the preferred method.

3.5.4.1.1 Signal detect indication

The SRDS_0/1_SIG_DET pins can be connected to a Signal Detect or loss-of-signal output that indicates when no laser light is being received when the 82576 is used in a 1000BASE-SX or -LX implementation (SerDes operation). It prevents false carrier cases occurring when transmission by a non connected port couples in to the input. Unfortunately, there is no standard polarity for this signal coming from different manufacturers. The CTRL.ILOS bit provides for inversion of the signal from different external SerDes vendors, and should be set when the external SerDes provides a negative-true loss-of-signal.

Note: This bit also inverts the LINK input that provides link status indication from the PHY (in GMII/MII mode) and thus should be set to 0 for proper internal PHY operation.

3.5.4.1.2 MAC Link Speed

SerDes operation is only defined for 1000 Mb/s operation. Other link speeds are not supported. When configured for the SerDes interface, the MAC speed-determination function is disabled and the Device Status register bits (STATUS.SPEED) indicate a value of 10b for 1000 Mb/s.

3.5.4.1.3 SerDes Mode Auto-Negotiation

In SerDes mode, after power up or the 82576 reset via GIO_PWR_GOOD, the 82576 initiates auto-negotiation based on the default settings in the device control and transmit configuration or PCS Link Control Word registers, as well as settings read from the EEPROM. If enabled in the EEPROM, the 82576 immediately performs auto-negotiation.

TBI mode auto-negotiation, as defined in clause 37 of the IEEE 802.3z standard, provides a protocol for two devices to advertise and negotiate a common operational mode across a GbE link. The 82576 fully supports the IEEE 802.3z auto-negotiation function when using the on-chip PCS and internal SerDes.

TBI mode auto-negotiation is used to determine the following information:

- Duplex resolution (even though the 82576 MAC only supports full-duplex in SerDes mode).
- Flow control configuration.

Note: Since speed for SerDes modes is fixed at 1000 Mb/s, speed settings in the Device Control register are unaffected by the auto-negotiation process.

Auto-negotiation can be initiated at power up or asserting GIO_PWR_GOOD by enabling specific bits in the EEPROM.

The auto-negotiation process is accomplished by the exchange of /C/ ordered sets that contain the capabilities defined in the PCS_ANADV register in the 3rd and 4th symbols of the ordered sets. Next page are supported using the PCS_NPTX_AN register.

Bits *FD* and *LU* in the Device Status (STATUS) register, and bits in the PCS_LSTS register provide status information regarding the negotiated link.

Auto-negotiation can be initiated by the following:

- LRST transition from 1b to 0b
- PCS_LCMD.AN_ENABLE transition from 0b to 1b
- Receipt of /C/ ordered set during normal operation
- Receipt of a different value of the /C/ ordered set during the negotiation process
- Transition from loss of synchronization to synchronized state (if AN_ENABLE is set).
- PCS_LCMD.AN_RESTART transition from 0b to 1b

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 *PCS_ANADV* fields 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. Table 3-30 lists the mapping of the *PCS_ANADV* fields to the Config_reg Base Page encoding per clause 37 of the standard.

Table 3-30. 802.3z Advertised Base Page Mapping

15	14	13:12	11:9	8:7	6	5	4:0
Nextp	Ack	RFLT	rsv	ASM	Hd	Fd	rsv

The partner advertisement can be seen in the PCS_ LPAB and PCS_ LPABNP registers.

3.5.4.1.4 Forcing Link

Forcing link can be accomplished by software by writing a 1b 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 or external SerDes).

Note: The *PCS_LCMD.AN_ENABLE* bit must be set to a logic zero to enable forcing link.

When link is forced via the *CTRL.SLU* bit, the link does not come up unless the LOS signal is asserted or an energy indication is received from the SerDes receiver, implying that there is a valid signal being received by the optics or the SerDes.

The source of the signal detect is fixed using bit *ENRGSRC* in the CONNSW register.

3.5.4.1.5 HW Detection of Non-Auto-Negotiation Partner

Hardware can detect a SerDes partner that sends idle code groups continuously, but does not initiate or answer an auto-negotiation process. In this case, hardware initiates an auto-negotiation process, and if it fails after some timeout, a link up is assumed. To enable this functionality the *PCS_LCTL.AN_TIMEOUT_EN* bit should be set. This mode can be used instead of the force link mode as a way to support a partner that do not support auto-negotiation.

3.5.4.2 SGMII Link Configuration

When working in SGMII mode, the actual link setting is done by the external PHY and is dependent on the settings of this PHY. The SGMII auto-negotiation process described in the sections that follow is only used to establish the MAC/PHY connection.

3.5.4.2.1 SGMII Auto-Negotiation

This auto-negotiation process is not dependent on the SRDS0/1_SIG_DET signal, as this signal indicates the status of the PHY signal detection (usually used in an optical PHY).

The outcome of this auto-negotiation process includes the following information:

- Link status
- Speed
- Duplex

This information is used by hardware to configure the MAC, when operating in SGMII mode.

Bits *FD* and *LU* of the Device Status (STATUS) register and bits in the PCS_LSTS register provide status information regarding the negotiated link.

Auto-negotiation can be initiated by the following:

- LRST transition from b1 to 0b.
- PCS_LCMD.AN_ENABLE transition from 0b to 1b.
- 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 a synchronized state (if AN_ENABLE is set).
- PCS_LCMD.AN_RESTART transition from 0b to 1b.

Resolving the negotiated link determines the 82576 operation with respect to speed and duplex settings. These negotiated capabilities override advertised and software controlled device configuration.

When working in SGMII mode, there is no need to set the PCAS_ANADV register, as the MAC advertisement word is fixed. The result of the SGMII level auto-negotiation can be read from the PCS_LPAB register.

3.5.4.2.2 Forcing Link

In SGMII, forcing of the link cannot be done at the PCS level, only in the external PHY. The forced speed and duplex settings are reflected by the SGMII auto-negotiation process; the MAC settings are automatically done according to this functionality.

3.5.4.2.3 MAC Speed Resolution

The MAC speed and duplex settings are always set according to the SGMII auto-negotiation process.

3.5.4.3 Copper PHY Link Configuration

When operating with the internal PHY, 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 designer desires to manually configure the link. The following sections discuss the methods of link configuration for copper PHY operation.

3.5.4.3.1 PHY Auto-Negotiation (Speed, Duplex, Flow Control)

When using a copper PHY, the PHY performs the auto-negotiation function. The actual operational details of this operation are described in the IEEE P802.3ab draft standard and are not included here.

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.

Note: If flow control is enabled in the 82576, the settings for the desired flow control behavior must be set by software in the PHY registers and auto-negotiation restarted. After auto-negotiation completes, the software device driver must read the PHY registers to determine the resolved flow control behavior of the link and reflect these in the MAC register settings (CTRL.TFCE and CTRL.RFCE).

Once PHY auto-negotiation completes, the PHY asserts a link indication (LINK) to the MAC. Software must have set the *Set Link Up* bit in the Device Control register (CTRL.SLU) before the MAC recognizes the LINK indication from the PHY and can consider the link to be up.

3.5.4.3.2 MAC Speed Resolution

For proper link operation, both the MAC and PHY must be configured for the same speed of link operation. The speed of the link can be determined and set by several methods with the 82576. These include:

- Software-forced configuration of the MAC speed setting based on PHY indications, which might be determined as follows:
  - — Software reads of PHY registers directly to determine the PHY's auto-negotiated speed
  - — Software reads the PHY's internal PHY-to-MAC speed indication (SPD_IND) using the MAC STATUS.SPEED register
- Software asks the MAC to attempt to auto-detect the PHY speed from the PHY-to-MAC RX_CLK, then programs the MAC speed accordingly
- MAC automatically detects and sets the link speed of the MAC based on PHY indications by using the PHY's internal PHY-to-MAC speed indication (SPD_IND)

Aspects of these methods are discussed in the sections that follow.

3.5.4.3.2.1 Forcing MAC Speed

There might be circumstances when the software device driver must forcibly set the link speed of the MAC. This can occur when the link is manually configured. To force the MAC speed, the software device driver must set the *CTRL.FRCSPD* (force-speed) bit to 1b and then write the speed bits in the Device Control register (*CTRL.SPEED*) to the desired speed setting. See Section 8.2.1 for details.

Note: Forcing the MAC speed using *CTRL.FRCSPD* overrides all other mechanisms for configuring the MAC speed and can yield non-functional links if the MAC and PHY are not operating at the same speed/configuration.

When forcing the 82576 to a specific speed configuration, the software device driver must also ensure the PHY is configured to a speed setting consistent with MAC speed settings. This implies that software must access the PHY registers to either force the PHY speed or to read the PHY status register bits that indicate link speed of the PHY.

Note: Forcing speed settings by *CTRL.SPEED* can also be accomplished by setting the *CTRL_EXT.SPD_BYPS* bit. This bit bypasses the MAC's internal clock switching logic and enables the software device driver complete control of when the speed setting takes place. The *CTRL.FRCSPD* bit uses the MAC's internal clock switching logic, which does delay the affect of the speed change.

3.5.4.3.2.2 Using Internal PHY Direct Link-Speed Indication

The 82576’s internal PHY provides a direct internal indication of its speed to the MAC (SPD_IND). When using the internal PHY, the most direct method for determining the PHY link speed and either manually or automatically configuring the MAC speed is based on these direct speed indications.

For MAC speed to be set/determined from these direct internal indications from the PHY, the MAC must be configured such that *CTRL.ASDE* and *CTRL.FRCSPD* are both 0b (both auto-speed detection and forced-speed override disabled). After configuring the Device Control register, MAC speed is re-configured automatically each time the PHY indicates a new link-up event to the MAC.

When MAC speed is neither forced nor auto-sensed by the MAC, the current MAC speed setting and the speed indicated by the PHY is reflected in the Device Status register bits *STATUS.SPEED*.

3.5.4.3.3 MAC Full-/Half- Duplex Resolution

The duplex configuration of the link is also resolved by the PHY during the auto-negotiation process. The 82576's internal PHY provides an internal indication to the MAC of the resolved duplex configuration using an internal full-duplex indication (FDX).

When using the internal PHY, this internal duplex indication is normally sampled by the MAC each time the PHY indicates the establishment of a good link (LINK indication). The PHY's indicated duplex configuration is applied in the MAC and reflected in the MAC Device Status register (*STATUS.FD*).

Software can override the duplex setting of the MAC 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 and the PHY's internal duplex indication is applied.

3.5.4.3.4 Using PHY Registers

The software device driver might be 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 (see Section 8.2.4). The MII registers enable the software device driver to have direct control over the PHY's operation, which can include:

- 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 set of PHY management registers required for all PHY devices can be found in the IEEE P802.3ab draft standard. The registers for the 82576 PHY are described in Section 3.5.8.

3.5.4.3.5 Comments Regarding Forcing Link

Forcing link in GMII/MII mode (internal PHY) requires the software device driver to configure both the MAC and PHY in a consistent manner with respect to each other as well as the link partner. After initialization, the software device driver configures the desired modes in the MAC, then accesses the PHY registers to set the PHY to the same configuration.

Before enabling the link, the speed and duplex settings of the MAC can be forced by software using the *CTRL.FRCSPD*, *CTRL.FRCDPX*, *CTRL.SPEED*, and *CTRL.FD* bits. After the PHY and MAC have both been configured, the software device driver should write a 1b to the *CTRL.SLU* bit.

3.5.4.4 Loss of Signal/Link Status Indication

For all modes of operation, an LOS/LINK signal provides an indication of physical link status to the MAC. When the MAC is configured for optical SerDes mode, the input reflects loss-of-signal connection from the optics. In backplane mode, where there is no LOS external indication, an internal indication from the SerDes receiver can be used. In SFP systems the LOS indication from the SFP can be used. In internal PHY 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).

When the link indication from the PHY is de-asserted or the loss-of-signal asserted from the SerDes, the MAC considers this to be a transition to a link-down situation (such as cable unplugged, loss of link partner, etc.). If the Link Status Change (LSC) interrupt is enabled, the MAC generates an interrupt to be serviced by the software device driver.

3.5.5 Ethernet Flow Control (FC)

The 82576 supports flow control as defined in 802.3x as well as the specific operation of asymmetrical flow control defined by 802.3z.

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 controlling 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.

The following registers are defined for the implementation of flow control:

- CTRL.RFCE field is used to enable reception of legacy flow control packets and reaction to them.
- CTRL.TFCE field is used to enable transmission of legacy flow control packets.
- Flow Control Address Low, High (FCAL/H) - 6-byte flow control multicast address
- Flow Control Type (FCT) 16-bit field to indicate flow control type
- Flow Control bits in Device Control (CTRL) register - Enables flow control modes.
- Discard PAUSE Frames (DPF) and Pass MAC Control Frames (PMCF) in RCTL - controls the forwarding of control packets to the host.
- Flow Control Receive Threshold High (FCRTH[1:0]) - A set of 13-bit high watermarks indicating receive buffer fullness. A single watermark is used in link FC mode.
- Flow Control Receive Threshold Low (FCRTL[1:0]) - A set of 13-bit low watermarks indicating receive buffer emptiness. A single watermark is used in link FC mode.
- Flow Control Transmit Timer Value (FCTTV) - a set of 16-bit timer values to include in transmitted PAUSE frame. A single timer is used in Link FC mode.
- Flow Control Refresh Threshold Value (FCRTV) - 16-bit PAUSE refresh threshold value

3.5.5.1 MAC Control Frames and Receiving Flow Control Packets

3.5.5.1.1 Structure of 802.3X FC 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 82576 (Receive Address Register 0).
2. A match on the type field
3. A comparison of the MAC *Control Op-Code* field

The 802.3x standard defines the MAC control frame multicast address as 01-80-C2-00-00-01.

The *Type* field in the FC packet is compared against an IEEE reserved value of 0x8808.

The final check for a valid PAUSE frame is the MAC control op-code. At this time only the PAUSE control frame op-code is defined. It has a value of 0x0001.

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 pause quantum (slot time). A pause quantum lasts 64 byte times, which is converted in absolute time duration according to the line speed.

Note: XON frame signals the cancellation of the pause from initiated by an XOFF frame - pause for zero pause quantum.

Table 3-31 lists the structure of a 802.3X FC packet

Table 3-31. 802.3X Packet Format

DA	01_80_C2_00_00_01 (6 bytes)
SA	Port MAC address (6 bytes)
Type	0x8808 (2 bytes)
Op-code	0x0001 (2 bytes)
Time	XXXX (2 bytes)
Pad	42 bytes
CRC	4 bytes

3.5.5.1.2 Operation and Rules

The 82576 operates in Link FC.

- Link FC is enabled by the RFCE bit in the CTRL Register.

Note: Link flow control capability is negotiated between link partners via the auto negotiation process. It is the software device driver responsibility to reconfigure the link flow control configuration after the capabilities to be used where negotiated as it might modify the value of these bits based on the resolved capability between the local device and the link partner.

Receiving a link FC frame while in PFC mode might be ignored. Receiving a PFC frame while in link FC mode is ignored.

Once the receiver has validated receiving an XOFF, or PAUSE frame, the 82576 performs the following:

- Increments the appropriate statistics register(s).
- Sets the *Flow_Control State* bit in the relevant FCSTS[0-1] register.
- Initializes the pause timer based on the packet's *PAUSE* timer field (overwriting any current timer's value).
- Disables packet transmission or schedules the disabling of transmission after the current packet completes.

Resumption of transmission might occur under the following conditions:

- Expiration of the PAUSE timer
- Reception of an XON frame (a frame with its PAUSE timer set to 0b)

Both conditions clear the relevant *Flow_Control State* bit in the relevant FCSTS[0-1] register and transmission can resume. Hardware records the number of received XON frames.

3.5.5.1.3 Timing Considerations

When operating at 1 Gb/s line speed, the 82576 must not begin to transmit a (new) frame more than two pause-quantum-bit times after receiving a valid link XOFF frame, as measured at the wires. A pause quantum is 512-bit times.

When operating in full duplex at 100 Mb/s or 1 Gb/s line speeds, the 82576 must not begin to transmit a (new) frame more than 576-bit times after receiving a valid link XOFF frame, as measured at the wire.

3.5.5.2 PAUSE and MAC Control Frames Forwarding

Two bits in the Receive Control register, control forwarding of PAUSE and MAC control frames to the host. These bits are *Discard PAUSE Frames* (*DPF*) and *Pass MAC Control Frames* (*PMCF*):

- The *DPF* bit controls forwarding of PAUSE packets to the host.
- The *PMCF* bit controls forwarding of non-PAUSE packets to the host.

Note: When virtualization is enabled, forwarded control packets are queued according to the regular switching procedure defined in Section 7.9.3.4.

When flow control reception is disabled (CTRL.RFCE = 0), flow control packets are not recognized and are parsed as regular packets.

Table 3-32 lists the behavior of the *DPF* bit.

Table 3-32. Forwarding of PAUSE Packet to Host (*DPF* Bit)

RFCE	DPF	Are FC Packets Forwarded to Host?
0	X	Yes. Packets needs to pass the L2 filters (see Section 7.1.2.1).[1]
1	0	Yes. Packets needs to pass the L2 filters (see Section 7.1.2.1).
1	1	No.

[1] The flow control multicast address is not part of the L2 filtering unless explicitly required.

Table 3-33 defines the behavior of the *PMCF* bit..

Table 3-33. Transfer of Non-PAUSE Control Packets to Host (*PMCF* Bit)

RFCE	PMCF	Are Non-FC MAC Control Packets Forwarded to Host?
0	X	Yes. Packets needs to pass the L2 filters (see Section 7.1.2.1).
X	0	Yes. Packets needs to pass the L2 filters (see Section 7.1.2.1).
1	1	Reserved.

3.5.5.3 Transmission of PAUSE Frames

The 82576 generates PAUSE packets to insure there is enough space in its receive packet buffers to avoid packet drop. The 82576 monitors the fullness of its receive packet buffers and compares it with the contents of a programmable threshold. When the threshold is reached, the 82576 sends a PAUSE frame. The 82576 also supports the sending of link Flow Control (FC).

Note: Similar to receiving link flow control packets previously mentioned, link XOFF packets can be transmitted only if this configuration has been negotiated between the link partners via the auto-negotiation process or some higher level protocol. The setting of this bit by the software device driver indicates the desired configuration.

The transmission of 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 82576 is operating in half-duplex mode.

3.5.5.3.1 Operation and Rules

Transmission of link PAUSE frames is enabled by software writing a 1b to the *TFCE* bit in the Device Control register.

The content of the Flow Control Receive Threshold High (FCRTH) register determines at what point the 82576 first transmits a PAUSE frame. The 82576 monitors the fullness of the receive packet buffer and compares it with the contents of FCRTH. When the threshold is reached, the 82576 sends a PAUSE frame with its pause time field equal to FCTTV.

At this time, the 82576 starts counting an internal shadow counter (reflecting the pause timeout counter at the partner end) from zero. When the counter reaches the value indicated in FCRTV register, then, if the PAUSE condition is still valid (meaning that the buffer fullness is still above the high watermark), an XOFF message is sent again.

Once the receive buffer fullness reaches the low water mark, the 82576 sends an XON message (a PAUSE frame with a timer value of zero). Software enables this capability with the *XONE* field of the FCRTL.

The 82576 sends a PAUSE frame if it has previously sent one and the packet buffer overflows. This is intended to minimize the amount of packets dropped if the first PAUSE frame did not reach its target. Since the secure receive packets use the same data path, the behavior is identical when secure packets are received.

3.5.5.3.2 Software Initiated PAUSE Frame Transmission

The 82576 has the added capability to transmit an XOFF frame via software. This is accomplished by software writing a 1b to the *SWXOFF* bit of the Transmit Control register. 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.

Note: The Flow Control Refresh Threshold mechanism does not work in the case of software-initiated flow control. Therefore, it is the software’s responsibility to re-generate PAUSE frames before expiration of the pause counter at the other partner's end.

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 0b in the PAUSE timer field of the FCTTV register. The software emission of XON packet is not allowed while the hardware flow control mechanism is active, as both use the FCTIV registers for different purposes.

XOFF transmission is not supported in 802.3x for half-duplex links. Software should not initiate an XOFF or XON transmission if the 82576 is configured for half-duplex operation.

When flow control is disabled, pause packets (XON, XOFF, and other FC) are not detected as flow control packets and can be counted in a variety of counters (such as multicast).

3.5.5.4 IPG Control and Pacing

The 82576 supports the following modes of controlling IPG duration:

- Fixed IPG - IPG is extended by a fixed duration
- Limiting payload rate - IPG is extended to limit the average data rate on the link.

3.5.5.4.1 Fixed IPG Extension

The 82576 allows controlling of the IPG duration. The IPGT configuration field enables an extension of IPG in 4-byte increments. One possible use of this capability is to allow the insertion of bytes into the transmit packet after it has been transmitted by the 82576 without violating the minimum IPG requirements. For example, a security device connected in series to the 82576 might add security headers to transmit packets before the packets go to the network.

3.5.5.4.2 Limiting Payload Rate

The 82576 allows controlling the maximum payload rate transmitted on the wire. Frames are spaced by an amount of idle time proportional to the maximum rate to achieve and to the length of the last frame transmitted. This feature is enabled by clearing bits *TCRSBYP* and *TCRSCOMP* in the RTTPCS register.

The maximum payload rate is defined for the entire link by setting the *RS_ENA* bit in the RTTPTCRC[0] register and by configuring RTTPTCRC[0] and RTTPTCRM[0] registers.

3.5.6 Loopback Support

3.5.6.1 General

The 82576 supports the following types of internal loopback in the LAN interfaces:

- MAC Loopback (Point 1 in figure)
- Internal PHY Loopback (Point 2 in figure)
- Internal SerDes Loopback (Point 3 in figure)
- External PHY Loopback (Point 4 in figure)

By setting the device to loopback mode, packets that are transmitted towards the line will be looped back to the host. The 82576 is fully functional in these modes, just not transmitting data over the lines. Figure 3-5 shows the points of loopback.

For more details on the usage and loopback test setup - See Intel® Ethernet Controllers Loopback Modes application note.

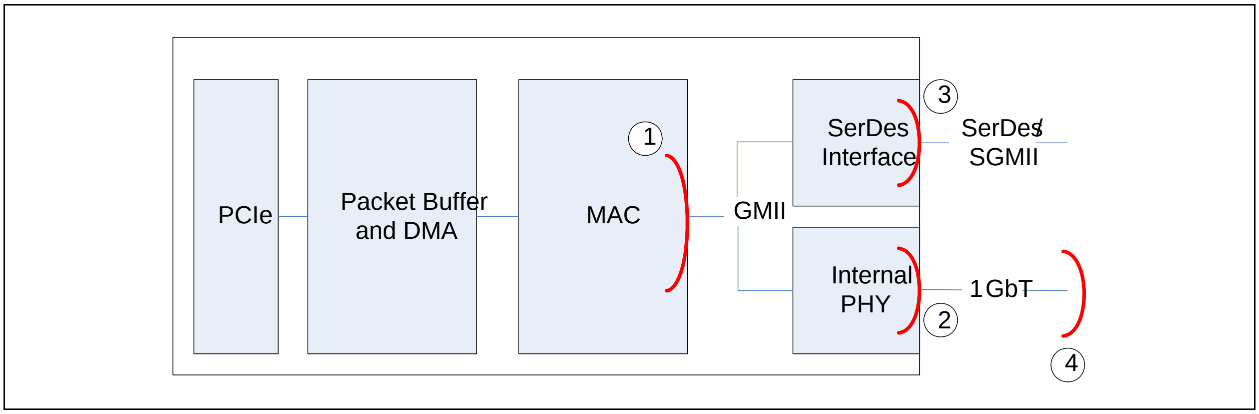


Figure 3-5. Intel® 82576 GbE Controller Loopback Modes

3.5.6.2 MAC Loopback

In MAC loopback, the PHY and SerDes blocks are not functional and data is looped back before these blocks. MAC loopback is operational only when working in PHY mode (CTRL_EXT.LINK_MODE = 00b).

3.5.6.2.1 Setting the 82576 to MAC loopback Mode

The following procedure should be used to put the 82576 in MAC loopback mode:

- Set RCTL.LBM to 2'b01 (bits 7:6)
- Set CTRL.SLU (bit 6, should be set by default)
- Set CTRL.FRCSPD & FRCDPLX (bits 11&12)
- Set CTRL.SPEED to 2'b10 (1G) and CTRL.FD
- Set CTRL.ILOS.
- Disable Auto negotiation in the PHY control register (Address 0 in the PHY):
  - — Clear Auto Neg enable bit (Bit 12)

Filter configuration and other TX/RX processes are as the same as I n normal mode.

Note: This configuration is for a case that there is no link in the PHY. If there is a link, ILOS bit should be cleared.

3.5.6.3 Internal PHY Loopback

In Internal PHY loopback the SerDes block is not functional and data is looped back at the end of the PHY functionality. This means all the design that is functional in copper mode, is involved in the loopback

3.5.6.3.1 Setting the 82576 to PHY loopback Mode

The following procedure should be used to put the 82576 in PHY loopback mode:

- Set Link mode to PHY: CTRL_EXT.LINK_MODE (CSR 0x18 BITS 23:22) = 0b00
- In PHY control register (Address 0 in the PHY):
  - — Set Duplex mode (bit 8)
  - — Set Loopback bit (Bit 14)
  - — Clear Auto Neg enable bit (Bit 12)
  - — Set speed using bits 6 and 13 as described in EAS.
  - — Register value should be:
    For 10 Mbps 0x4100
    For 100 Mbps 0x6100
    For 1000 Mbps 0x4140.
- In port control register (Address 16 (0x10) in the PHY), set bit 14 (Link disable). This is not a must for 1G but required for 10/100Mbps

While in loopback mode, polling for link might not return a valid link state. Transmit and receive normally.

Note: Make sure a Configure command is re-issued (loopback bits set to 00b) to cancel the loopback mode.

3.5.6.4 SerDes Loopback

In SerDes loopback the PHY block is not functional and data is looped back at the end of the SerDes functionality. This means all the design that is functional in SerDes/SGMII mode, is involved in the loopback .

Note: SerDes loopback is functional only if the SerDes link is up.

3.5.6.4.1 Setting the 82576 to SerDes loopback Mode

The following procedure should be used to put the 82576 in SerDes loopback mode:

- Set Link mode to SerDes: CTRL_EXT.LINK_MODE (CSR 0x18 BITS 23:22) = 0b11
- Configure SERDES (register 4 bit 1) to loopback: write to SERDESCTL (CSR 0x00024) the value 0x410
- Move to Force mode by setting the following bits:
    - — CTRL.FD (CSR 0x0 bit 0) = 1
    - — CTRL.SLU (CSR 0x0 bit 6) = 1
    - — CTRL.RFCE (CSR 0x0 bit 27) = 0
    - — CTRL.TFCE (CSR 0x0 bit 28) = 0
    - — CTRL.ILOS (CSR 0x0 bit 7) = 1
    - — CTRL.LRST (CSR 0x0 bit 3) = 0
    - — PCS_LCTL.FORCE_LINK (CSR 0X04208 bit 5) = 1
    - — PCS_LCTL.FSD (CSR 0X04208 bit 4) = 1
    - — PCS_LCTL.FDV (CSR 0X04208 bit 3) = 1
    - — PCS_LCTL.FLV (CSR 0X04208 bit 0) = 1
    - — PCS_LCTL.AN_ENABLE (CSR 0X04208 bit 16) = 0

3.5.6.5 External PHY Loopback

In External PHY loopback the SerDes block is not functional and data is sent through the MDI interface and looped back using an external loopback plug. This means all the design that is functional in copper mode, is involved in the loopback.

3.5.6.5.1 Setting the 82576 to External PHY loopback Mode

The following procedure should be used to put the 82576 in PHY loopback mode:

- Set Link mode to PHY: CTRL_EXT.LINK_MODE (CSR 0x18 BITS 23:22) = 0b00
- In PHY control register (Address 0 in the PHY):
    - — Set Duplex mode (bit 8)
    - — Clear Loopback bit (Bit 14)
    - — Set Auto Neg enable bit (Bit 12)
- Tune the PHY DSP to Loopback operation (in 1 Gbps mode only) using the following sequence:
    - — Set PHY Register address 0x12 to 0x1610 (Set non-flip-chip, auto mdix and enable gig loopback on TP)
    - — Turn off NEXT cancellers using the following command:
        - Set PHY Register address 0x1f37 to 0x3f1c.
    - — Wait 10 milliseconds
    - — Set PHY Register address 0x9 to 0x1a00
    - — Set PHY Register address 0x12 to 0xC10
    - — Set PHY Register address 0x12 to 0x1C10
    - — Set PHY Register address 0x1F35 to 0x65

- — Set PHY Register address 0x1837 to 0x3F7C
- — Set PHY Register address 0x1437 to 0x3FDC
- — Set PHY Register address 0x1237 to 0x3F7C
- — Set PHY Register address 0x1137 to 0x3FDC

- Restart auto-negotiation (Set bit 9)
- Reset the PHY (Set bit 15).
- Wait for auto-negotiation to complete, then transmit and receive normally.

3.5.7 Integrated Copper PHY Functionality

The PHY default configuration is determined by data from the EEPROM, which is read right after power-on reset.

The register set used to control the PHY functionality (PHYREG) is described in Section 8.24.

3.5.7.1 PHY Initialization Functionality

3.5.7.1.1 Auto MDIO Register Initialization

The 82576 PHY supports an option to automatically initialize MDIO registers with values from EEPROM/ ROM if the hardware defaults are not adequate.

In the 82576, this is performed by the MMS unit (firmware).

There are two types of register initialization:

1. General register initialization - any register in PHY can be initialized.
2. EEPROM bit initialization - there are some bits in the PHY that are a mirror of EEPROM bit - 25.6, 25.3:0, 26.0.

After any PHY reset (power down included), the PHY needs to be initialized for both steps 1 and 2.

The register initialization is done by the MMS (firmware) through the MAC/PHY MDIO interface (MDIC).

3.5.7.1.2 General Register Initialization

A block of data is allocated in EEPROM/ROM (see Section 6.4).

This block holds register addresses and data in MDIC format (Section 8.2.4).

Every time a PHY reset ends, this block is read from EEPROM by the MMS and is written to PHY registers through the MDIC registers and the MDIO interface.

3.5.7.1.3 Mirror Bit Initialization

There are a number of bits (OEM bits) that reside in EEPROM/MAC control registers that have a mirror bit in the PHY registers. These bits are also updated by the MMS after every PHY reset.

These bits are updated after the general register initialization and through a read modify write sequence.

The current mirror bits are: registers - 25.6, 25.3:0, and 26.0.

3.5.7.2 Determining Link State

The PHY and its link partner determine the type of link established through one of three methods:

- Auto-negotiation
- Parallel detection
- Forced operation

Auto-negotiation is the only method allowed by the 802.3ab standard for establishing a 1000BASE-T link, although forced operation could be used for test purposes. For 10/100 links, any of the three methods can be used. The following sections discuss each in greater detail.

Figure 3-6 provides an overview of link establishment. First the PHY checks if auto-negotiation is enabled. By default, the PHY supports auto-negotiation, see PHY Register 0, bit 12. If not, the PHY forces operation as directed. If auto-negotiation is enabled, the PHY begins transmitting Fast Link Pulses (FLPs) and receiving FLPs from its link partner. If FLPs are received by the PHY, auto-negotiation proceeds. It also can receive 100BASE-TX MLT3 and 10BASE-T Normal Link Pulses (NLPs). If either MLT3 or NLPs are received, it aborts FLP transmission and immediately brings up the corresponding half-duplex link.

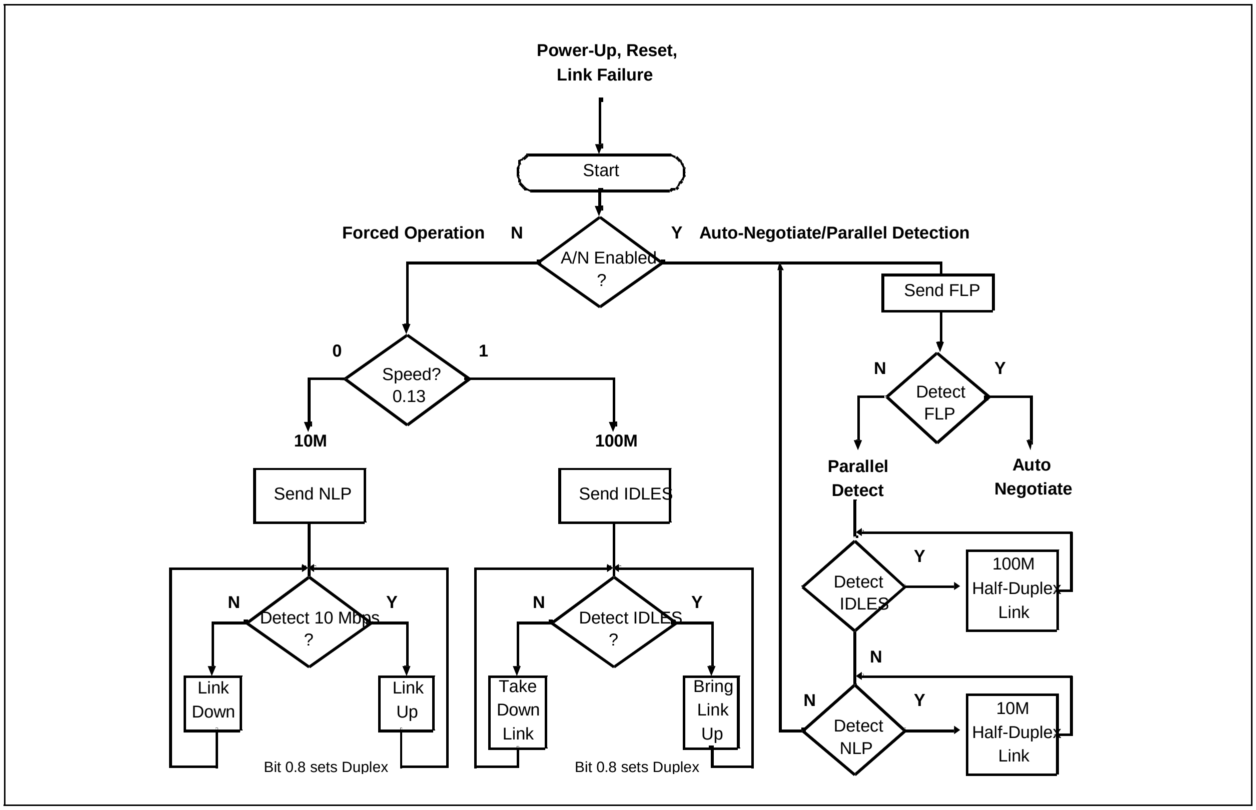


Figure 3-6. Overview of Link Establishment

3.5.7.2.1 False Link

The PHY does not falsely establish link with a partner operating at a different speed. For example, the PHY does not establish a 1 Gb/s or 10 Mb/s link with a 100 MB/s link partner.

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, designers 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
  - — 10BASE-T, 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.

3.5.7.2.2 Forced Operation

Forced operation can be used to establish 10 Mb/s and 100 Mb/s links, and 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 MII Register 0.

Note: When speed is forced, the auto cross-over feature is not functional.

In forced operation, the designer sets the link speed (10, 100, or 1000 MB/s) and duplex state (full or half). For Gigabit (1000 MB/s) links, designers must explicitly designate one side as the master and the other as the slave.

Note: The paradox (per the standard): If one side of the link is forced to full-duplex operation and the other side has auto-negotiation enabled, the auto-negotiating partner parallel-detects to a half-duplex link while the forced side operates as directed in full-duplex mode. The result is spurious, unexpected collisions on the side configured to auto-negotiate.

Table 3-34 lists link establishment procedures.

Table 3-34. 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.).

3.5.7.2.3 Auto Negotiation

The PHY supports the IEEE 802.3u auto-negotiation scheme with next page capability. Next page exchange uses Register 7 to send information and Register 8 to receive them. Next page exchange can only occur if both ends of the link advertise their ability to exchange next pages.

3.5.7.2.4 Parallel Detection

Parallel detection can only be used to establish 10 and 100 Mb/s links. It occurs when the PHY tries to negotiate (transmit FLPs to its link partner), but instead of sensing FLPs from the link partner, it senses 100BASE-TX MLT3 code or 10BASE-T Normal Link Pulses (NLPs) instead. In this case, the PHY immediately stops auto-negotiation (terminates transmission of FLPs) and immediately brings up whatever link corresponds to what it has sensed (MLT3 or NLPs). If the PHY senses both technologies, the 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 the master/slave relationship required by 1000BASE-T. This is why parallel detection cannot be used to establish GbE links.

3.5.7.2.5 Auto Cross-Over

Twisted pair Ethernet PHY's must be correctly configured for MDI or MDI-X operation to inter operate. This has historically been accomplished using special patch cables, magnetics pinouts or Printed Circuit Board (PCB) wiring. The PHY supports the automatic MDI/MDI-X configuration originally developed for 1000Base-T and standardized in IEEE 802.3u section 40. Manual (non-automatic) configuration is still possible.

For 1000BASE-T links, pair identification is determined automatically in accordance with the standard.

For 10/100 Mb/s inks and during auto-negotiation, pair usage is determined by bits 12 and 13 in the Port Control Register (PHYREG18).

In addition, the PHY has an automatic cross-over detection function. If bit 18.12 = 1b, the PHY automatically detects which application is being used and configures itself accordingly.

The automatic MDI/MDI-X state machine facilitates switching the MDI_PLUS[0] and MDI_MINUS[0] signals with the MDI_PLUS[1] and MDI_MINUS[1] signals, respectively, prior to the auto-negotiation mode of operation so that FLPs can be transmitted and received in compliance with Clause 28 auto-negotiation specifications. An algorithm that controls the switching function determines the correct polarization of the cross-over circuit. This algorithm uses an 11-Bit Linear Feedback Shift Register (LFSR) to create a pseudo-random sequence that each end of the link uses to determine its proposed configuration. After making the selection to either MDI or MDI-X, the node waits for a specified amount of time while evaluating its receive channel to determine whether the other end of the link is sending link pulses or PHY-dependent data. If link pulses or PHY-dependent data are detected, it remains in that configuration. If link pulses or PHY-dependent data are not detected, it increments its LFSR and makes a decision to switch based on the value of the next bit. The state machine does not move from one state to another while link pulses are being transmitted.

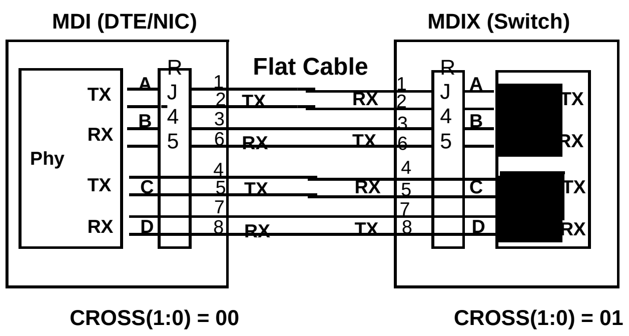


Figure 3-7. Cross-Over Function

3.5.7.2.6 10/100 MB/s Mismatch Resolution

It is a common occurrence that a link partner (such as a switch) is configured for forced full-duplex 10/ 100 Mb/s operation. The normal auto-negotiation sequence would result in the other end settling for half-duplex 10/100 Mb/s operation. The mechanism described in this section resolves the mismatch and automatically transitions the 82576 into FDX mode, enabling it to operate with a partner configured for FDX operation.

The 82576 enables the system software device driver to detect the mismatch event previously described and sets its duplex mode to the appropriate value without a need to go through another auto-negotiation sequence or breaking link. Once software detects a possible mismatch, it might instruct the 82576 to change its duplex setting to either HDX or FDX mode. Software sets the *Duplex_manual_set* bit to indicate that duplex setting should be changed to the value indicated by the *Duplex Mode* bit in PHY Register 0. Any change in the value of the *Duplex Mode* bit in PHY Register 0 while the *Duplex_manual_set* bit is set to 1b would also cause a change in the device duplex setting.

The *Duplex_manual_set* bit is cleared on all PHY resets, following auto-negotiation, and when the link goes down. Software might track the change in duplex through the PHY *Duplex Mode* bit in Register 17 or a MAC indication.

3.5.7.2.7 Link Criteria

Once the link state is determined-via auto-negotiation, parallel detection or forced operation, the PHY and its link partner bring up the link.

3.5.7.2.7.1 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. These coefficients are used to equalize the incoming signal, as well as eliminate signal impairments such as echo and cross talk.

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 each time it has no data to transmit. The link is maintained as long as valid idle, data, or carrier extension symbols are received.

3.5.7.2.7.2 100BASE-TX

For 100BASE-TX links, the PHY and its link partner immediately begin transmitting idle symbols. Each side continues sending idle symbols each time 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 each time the scrambler becomes locked and remains locked for approximately 50 ms. Link remains up unless the de scrambler receives less than 12 consecutive idle symbols in any 2 ms period. This provides for a very robust operation, essentially filtering out any small noise hits that might otherwise disrupt the link.

3.5.7.2.7.3 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.

In 10 Mb/s mode, the PHY establishes link based on the link state machine found in 802.3, clause 14.

Note: 100 Mb/s idle patterns do not bring up a 10 Mb/s link.

3.5.7.3 Link Enhancements

The PHY offers two enhanced link functions, each of which are discussed in the sections that follow:

- SmartSpeed
- Flow control

3.5.7.3.1 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 allow 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 a configurable number (1-5, Register 27.8:6) of failed attempts, the PHY automatically downgrades the highest ability it advertises to the next lower speed: from 1000 to 100 to 10 Mb/s. Once a link is established, and if it is later broken, the PHY automatically upgrades the capabilities advertised to the original setting. This enables the PHY to automatically recover once the cable plant is repaired.

3.5.7.3.1.1 Using SmartSpeed

SmartSpeed is enabled by setting PHYREG.16.7 = 1b. When SmartSpeed downgrades the PHY advertised capabilities, it sets bit PHYREG.19.5. When link is established, its speed is indicated in PHYREG.17.15:14. SmartSpeed automatically resets the highest-level auto-negotiation abilities advertised, if link is established and then lost for more than 2 seconds.

The number of failed attempts allowed is configured by Register 27.8:6.

Note: SmartSpeed and M/S fault - When SmartSpeed is enabled, the M/S (Master-Slave) resolution is not given seven attempts to try to resolve M/S status (see IEEE 802.3 clause 40.5.2), this is due to the fact that SmartSpeed downgrades the link after at most five attempts.

Time To Link with Smart Speed - in most cases, any attempt duration is approximately 2.5 seconds, in other cases it could take more than 2.5 seconds depending on configuration and other factors.

3.5.7.4 Flow Control

Flow control is a function that is described in Clause 31 of the IEEE 802.3 standard. It allows congested nodes to pause traffic. Flow control is essentially a MAC-to-MAC function. MACs indicate their ability to implement flow control during auto-negotiation. This ability is communicated through two bits in the auto-negotiation registers (PHYREG.4.10 and PHYREG.4.11).

The PHY transparently supports MAC-to-MAC advertisement of flow control through its auto-negotiation process. Prior to auto-negotiation, the MAC indicates its flow control capabilities via PHYREG.4.10 (Pause) and PHYREG.4.11 (ASM_DIR). After auto-negotiation, the link partner's flow control capabilities are indicated in PHYREG.5.10 and PHYREG.5.11.

There are two forms of flow control that can be established via auto-negotiation: symmetric and asymmetric. Symmetric flow control is for point-to-point links; asymmetric for hub-to-end-node connections. Symmetric flow control enables either node to flow-control the other. Asymmetric flow-control enables a repeater or switch to flow-control a DTE, but not vice versa.

Table 3-35 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 3-35. Pause And Asymmetric Pause Settings

ASM_DIR settings Local (PHYREG.4.10) and Remote (PHYREG.5.10)	Pause Setting - Local (PHYREG.4.9)	Pause Setting - Remote (PHYREG.5.9)	Result
Both ASM_DIR = 1b	1	1	Symmetric - Either side can flow control the other
	1	0	Asymmetric - Remote can flow control local only
	0	1	Asymmetric - Local can flow control remote
	0	0	No flow control
Either or both ASM_DIR = 0b	1	1	Symmetric - Either side can flow control the other
	Either or both = 0		No flow control

3.5.7.5 Management Data Interface

The PHY supports the IEEE 802.3 MII Management Interface also known as the Management Data Input/Output (MDIO) Interface. This interface enables upper-layer devices to monitor and control the state of the PHY. The MDIO interface consists of a physical connection, a specific protocol that runs across the connection, and an internal set of addressable registers.

The PHY supports the core 16-bit MDIO registers. Registers 0-10 and 15 are required and their functions are specified by the IEEE 802.3 specification. Additional registers are included for expanded functionality. Specific bits in the registers are referenced using an PHY REG X.Y notation, where X is the register number (0-31) and Y is the bit number (0-15). See the software interface chapter.

3.5.7.6 Low Power Operation and Power Management

The PHY incorporates numerous features to maintain the lowest power possible.

The PHY can be entered into a low-power state according to MAC control (Power Management controls) or via PHY Register 0. In either power down mode, the PHY is not capable of receiving or transmitting packets.

3.5.7.6.1 Power Down via the PHY Register

The PHY can be powered down using the control bit found in PHYREG.0.11. 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 register power down. The power down bit is active high. When the PHY exits software power-down (PHYREG.0.11 = 0b), it re-initializes all analog functions, but retains its previous configuration settings.

3.5.7.6.2 Power Management State

PHY is aware of power management state. If the PHY is not in a power down state, then PHY behavior regarding several features are different depending on the power state. See Section 3.5.7.6.4 for details.

The internal PHY power state can be accessed (and overridden) through Register 27.2:0.

3.5.7.6.3 AN1000_dis

AN1000_dis is an option to disable 1000 Mb/s advertisement in PHY regardless of Register 9.

This is for cases where the system doesn't support working in 1000 Mb/s due to power limitations.

This option is enabled by following bits in PHY registers:

- PHYREG 25.3 - disable 1000 Mb/s when in non-D0a states only.
- PHYREG 25.6 - disable 1000 Mb/s always.
- PHYREG 26.0 - same as 25.6, but this is a secure bit (see Secure Register chapter).

3.5.7.6.4 Low Power Link Up - Link Speed Control

Normal PHY speed negotiation drives to establish a link at the highest possible speed. PHY supports an additional mode of operation, where the PHY drives to establish a link at a low speed. The link-up process enables 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 previously described should be disabled (in both D0a state and non-D0a states) when the designer advertisement is anything other than 10/100/1000 Mb/s (all three). This is to avoid reaching (through the LPLU procedure) a link speed that is not advertised by the user.

Table 3-36 lists link speed as function of power management state, link speed control, and GbE speed enabling:

Table 3-36. Link Speed vs. Power State

Power Management State	Low Power Link Up (reg 25.1 and 2)	GbE Disable Bits		PHY Speed Negotiation
		Disable 1000 (reg 25.6)	Disable 1000 in non-D0a (reg 25.3)	
D0a	0, Xb	0b	X	PHY negotiates to highest speed advertised (normal operation).
		1b		PHY negotiates to highest speed advertised (normal operation), excluding 1000 Mb/s.
	1, Xb	0b	X	PHY goes through Low Power Link Up (LPLU) procedure, starting with advertised values.
		1b		PHY goes through LPLU procedure, starting with advertised values. Does not advertise 1000 Mb/s.

Table 3-36. Link Speed vs. Power State

Non-D0a	X, 0b	0b	0b	PHY negotiates to highest speed advertised.
		0b	1b	PHY negotiates to highest speed advertised, excluding 1000 Mb/s.
		1b	X	
	X, 1b	0b	0b	PHY goes through LPLU procedure, starting at 10 Mb/s.
		0b	1b	PHY goes through LPLU procedure, starting at 10 Mb/s. Does not advertise 1000 Mb/s.

The PHY initiates auto-negotiation without a direct driver command in the following cases:

- When the state of *Disable_1000* changes. For example, if 1000 Mb/s is disabled on D3 or Dr entry (but not in D0a), the PHY auto-negotiates on entry.
- When LPLU changes state with a change in a power management state. For example, on transition from D0a without LPLU to D3 with LPLU. Or, on transition from D3 with LPLU to D0 without 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.

3.5.7.6.4.1 D0a State

A power-managed link speed control lowers link speed (and power) when highest link performance is not required. When enabled (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 D0LPLU configuration bit enables *D0 Low Power Link Up*. Before enabling this feature, software must advertise to one of the following speed combinations: 10 Mb/s only, 10/100 Mb/s only, or 10/100/1000 Mb/s.

When speed negotiation starts, the PHY tries to negotiate at a speed based on the currently advertised values. If link establishment fails, the PHY tries to negotiate with different speeds; it enables all speeds up to the lowest speed supported by the partner. For example, PHY advertises 10 Mb/s only, and the partner supports 1000 Mb/s only. After the first try fails, the PHY 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 detect, if needed. Automatic MDI/MDI-X resolution is done during the first auto-negotiation stage.

3.5.7.6.4.2 Non-D0a State

The PHY might negotiate to a low speed while in non-D0a states (Dr, D0u, D3). This applies only when the link is required by one of the following: SMBus manageability, APM Wake, or PME. Otherwise, the PHY is disabled during the non-D0 state.

The *Low Power on Link-Up* (Register 25.2, is also loaded from EEPROM) bit enables reduction in link speed:

- At power-up entry to Dr state, the PHY advertises supports for 10 Mb/s only and goes through the link up process.
- At any entry to a non-D0a state (Dr, D0u, D3), the PHY advertises support for 10 Mb/s only and goes through the link up process.
- 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.

Link negotiation begins with the PHY trying to negotiate at 10 Mb/s speed only regardless of user auto-negotiation advertisement. If link establishment fails, the PHY tries to negotiate at additional speeds; it enables all speeds up to the lowest speed supported by the partner. For example, the PHY advertises 10 Mb/s only and the partner supports 1000 Mb/s only. After the first try fails, PHY 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 detect, if needed. Automatic MDI/MDI-X resolution is done during the first auto-negotiation stage.

3.5.7.6.5 Smart Power-Down

Smart power-down is a link-disconnect capability applicable to all power management states. Smart Power-down combines a power saving mechanism with the fact that the link might disappear and resume.

Smart power-down is enabled by PHYREG 25.0 or by *SPD Enable* bit in the EEPROM and is entered when the PHY detects link loss. Auto-negotiation must also be enabled. While in the smart power-down state, the PHY powers down circuits and clocks that are not required for detection of link activity. The PHY is still be able to detect link pulses (including parallel detect) and wakeup to engage in link negotiation. The PHY does not send link pulses (NLP) while in SPD state; however, register accesses are still possible.

When the PHY is in smart power-down and detects link activity, it re-negotiates link speed based on the power state and the *Low Power Link Up* bit as described in PHYREG 25.1 and 25.2.

Note: The link-disconnect state applies to all power management states (Dr, D0u, D0a, D3).

The link might change status, that is go up or go down, while in any of these states.

3.5.7.6.5.1 Back-to-Back Smart Power-Down

While in link disconnect, the 82576 monitors the link for link pulses to identify when a link is re-connected. The 82576 also periodically transmits pulses to resolve the case of two the 82576s (or devices with the 82576-like behavior) connected to each other across the link. Otherwise, two such devices might be locked in Smart power-down mode, not capable of identifying that a link was re-connected.

The link pulses are transmitted on average every 100 ms on alternate channels (A/B and C/D) and add <1% to total the 82576 power in link disconnect mode. Pulses do not conform to IEEE specification regarding link pulse template. A single pulse should be enough to bring a receiver out of smart power-down mode in a worst-case configuration (such as maximum cable length, highest cable attenuation, etc.).

If the link partners are disconnected and then reconnected, it is possible that the two controllers transmit their pulses at the same time. Since the 82576 masks its receiver during pulse transmission, such synchronization causes pulses to be missed by both partners. A randomization factor is therefore applied to the timing of transmitted pulses, affecting the period between pulses. The randomization factor is specific per device and should reduce the probability of a lock to 10^{-4}. Note that if the two partners happen to transmit within the same slot, and if the randomization factor happens to be similar, it takes longer for the partners to get out of sync with each other.

Back-to-back smart power-down is enabled by the *SPD_B2B_EN* bit in the PHY registers. The default value is enabled. The *Enable* bit applies to smart power-down mode.

Note: This bit should not be altered by software once the 82576 was set in smart power-down mode. If software requires changing the back-to-back status, it first needs to transition the PHY out of smart power-down mode and only then change the back-to-back bit to the required state.

3.5.7.6.6 Link Energy Detect

The PHY asserts the *Link Energy Detect* bit (PHYREG 25.4) each time energy is detected on the link. This bit provides an indication of a cable becoming plugged or unplugged.

This bit is valid only if auto-negotiation is enabled and smart power-down is enabled (reg 25.0).

In order to correctly deduce that there is no energy, the bit must read 0b for three consecutive reads each second.

3.5.7.6.7 PHY Power-Down State

Each 82576 port enters a power-down state when none of its clients is enabled and therefore has no need to maintain a link. This can happen in one of the following cases. Note that PHY power-down must be enabled through the EEPROM PHY *Power Down Enable* bit.

1. D3/Dr state: Each PHY enters a low-power state if the following conditions are met:
    a. The LAN function associated with this PHY is in a non-D0 state
    b. APM WOL is inactive
    c. Manageability doesn't use this port.
    d. ACPI PME is disabled for this port.
    e. The PHY *Power Down Enable* EEPROM bit is set (word 0xF, bit 6).
2. SerDes mode: Each PHY is disabled when its LAN function is configured to SerDes mode.
3. LAN disable: Each PHY can be disabled if its LAN function's LAN Disable input indicates that the relevant function should be disabled. Since the PHY is shared between the LAN function and manageability, it might not be desirable to power down the PHY in LAN Disable. The *PHY_in_LAN_Disable* EEPROM bit determines whether the PHY (and MAC) are powered down when the LAN Disable pin is asserted. The default is not to power down.

A LAN port can also be disabled through EEPROM settings. If the *LAN_DIS* EEPROM bit is set, the PHY enters power down. Note, however, that setting the EEPROM *LAN_PCI_DIS* bit does not bring the PHY into power down.

3.5.7.7 Advanced Diagnostics

The 82576 PHY incorporates hardware support for advanced diagnostics.

The hardware support enables output of internal PHY data to host memory for post processing by the software device driver.

Diagnostics supported are:

3.5.7.7.1 TDR - Time Domain Reflectometry

By sending a pulse onto the twisted pair and observing the retuned signal, the following can be deduced:

1. Is there a short?
2. Is there an open?
3. Is there an impedance mismatch?
4. What is the length to any of these faults?

3.5.7.7.2 Channel Frequency Response

By doing analysis on the Tx and Rx data, it can be established that a channel's frequency response (also known as insertion loss) can determine if the channel is within specification limits. (Clause 40.7.2.1 in IEEE 802.3).

3.5.7.8 1000 Mb/s Operation

3.5.7.8.1 Introduction

Figure 3-8 shows an overview of 1000BASE-T functions, followed by discussion and review of the internal functional blocks.

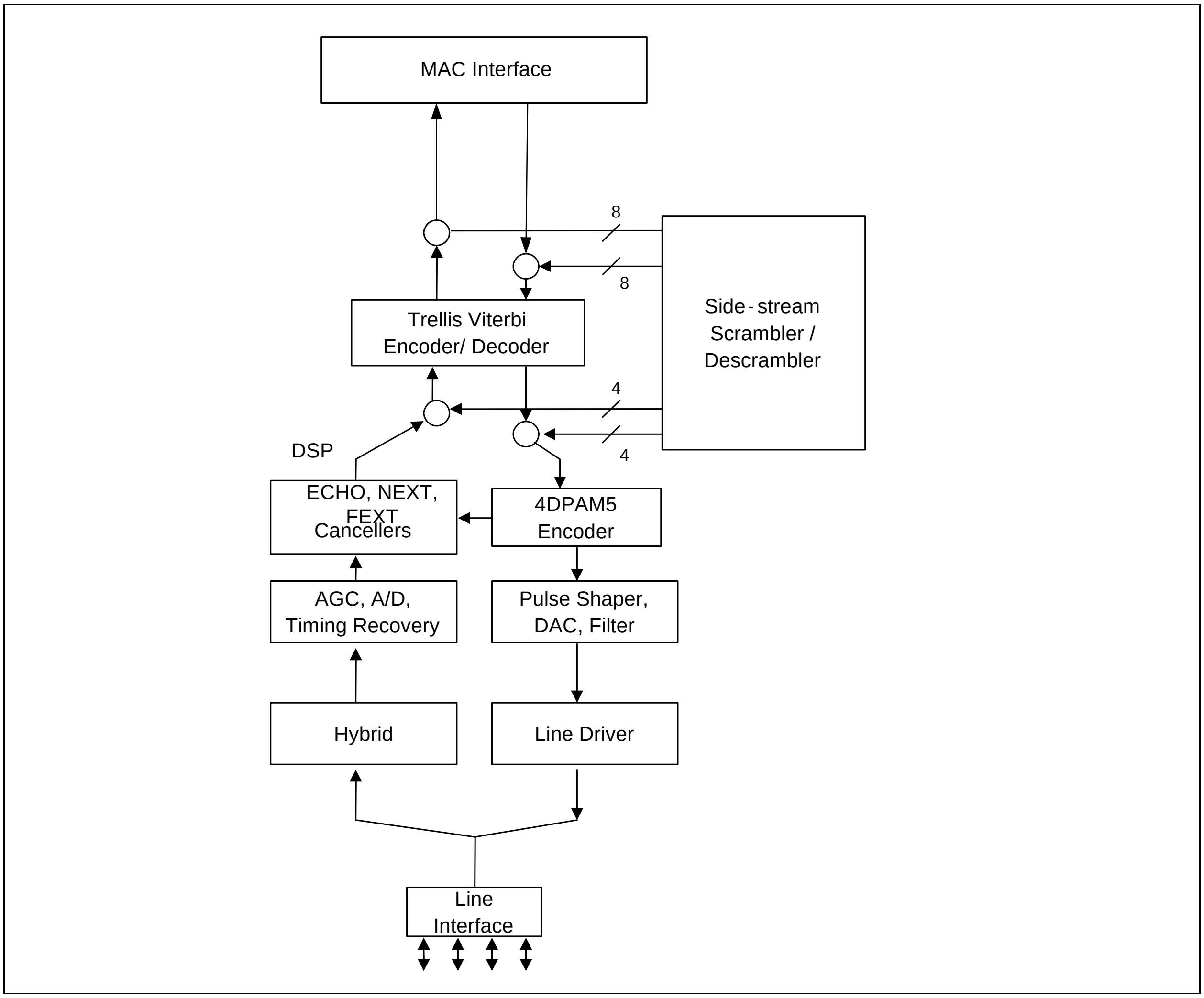


Figure 3-8. 1000BASE-T Functions Overview

3.5.7.8.2 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 (see Figure 3-8).

3.5.7.8.2.1 Scrambler

The scrambler randomizes the transmitted data. The purpose of scrambling is twofold:

1. Scrambling eliminates repeating data patterns (also known as spectral lines) from the 4DPAM5 waveform in order to reduce EMI.
2. Each channel (A, B, C, D) has a unique signature that the receiver uses for identification.

The scrambler is driven by a 33-bit 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 twelve mutually uncorrelated outputs. Eight of these are used to randomize the inputs to the 4DPAM5 and Trellis encoders. The remaining four outputs randomize the sign of the 4DPAM5 outputs.

3.5.7.8.2.2 Transmit FIFO

The transmit FIFO re-synchronizes data transmitted by the MAC to the transmit reference used by the PHY. The FIFO is large enough to support a frequency differential of up to +/- 1000 ppm over a packet size of 10 KB (jumbo frame).

3.5.7.8.2.3 Transmit Phase-Locked Loop PLL

This function generates the 125 MHz timing reference used by the PHY to transmit 4DPAM5 symbols. When the PHY is the master side of the link, the XI input is the reference for the transmit PLL. When the PHY is the slave side of the link, the recovered receive clock is the reference for the transmit PLL.

3.5.7.8.2.4 Trellis Encoder

The Trellis encoder uses the two high-order bits of data and its previous output to generate a ninth bit, which determines if the next 4DPAM5 pattern should be even or odd.

For data, this function is:

```
Trellisn = Data7n-1 XOR Data6n-2 XOR Trellisn-3
```

This provides forward error correction and enhances the Signal-To-Noise (SNR) ratio by a factor of 6 dB.

3.5.7.8.2.5 4DPAM5 Encoder

The 4DPAM5 encoder translates 8-byte codes transmitted by the MAC into 4DPAM5 symbols. The encoder operates at 125 MHz, which is both the frequency of the MAC interface and the baud rate used by 1000BASE-T.

Each 8-byte code represents one of 28 or 256 data patterns. Each 4DPAM5 symbol consists of one of five signal levels (-2,-1,0,1,2) on each of the four twisted pair (A,B,C,D) representing 54 or 625 possible patterns per baud period. Of these, 113 patterns are reserved for control codes, leaving 512 patterns for data. These data patterns are divided into two groups of 256 even and 256 odd data patterns. Thus, each 8-byte octet has two possible 4DPAM5 representations: one even and one odd pattern.

3.5.7.8.2.6 Spectral Shaper

This function causes the 4DPAM5 waveform to have a spectral signature that is very close to that of the MLT3 waveform used by 100BASE-TX. This enables 1000BASE-T to take advantage of infrastructure (cables, magnetics) designed for 100BASE-TX.

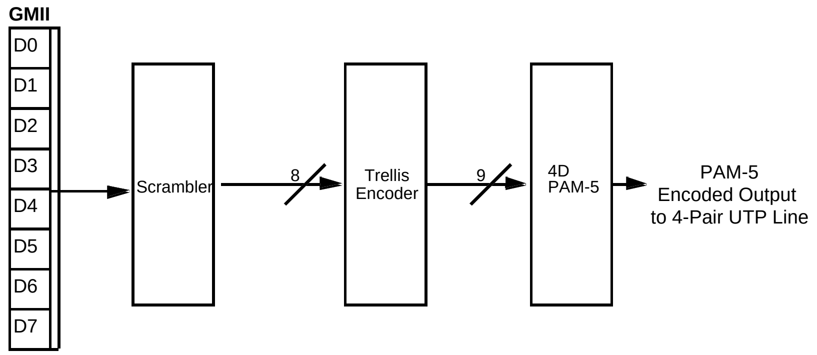
The shaper works by transmitting 75% of a 4DPAM5 code in the current baud period, and adding the remaining 25% into the next baud period.

3.5.7.8.2.7 Low-Pass Filter

To aid with EMI, this filter attenuates signal components more than 180 MHz. In 1000BASE-T, the fundamental symbol rate is 125 MHz.

3.5.7.8.2.8 Line Driver

The line driver drives the 4DPAM5 waveforms onto the four twisted-pair channels (A, B, C, D), adding them onto the waveforms that are simultaneously being received from the link partner.



Scrambler Polynomials:

$1 + x^{13} + x^{33}$ (Master PHY Mode)

$1 + x^{20} + x^{33}$ (Slave PHY Mode)

Figure 3-9. 1000BASE-T Transmit Flow And Line Coding Scheme

3.5.7.8.3 Receive Functions

This section describes function blocks that are used when the PHY receives data from the twisted pair interface and passes it back to the MAC (see Figure 3-10).

Figure 3-10. Transmit/Receive Flow

3.5.7.8.3.1 Hybrid

The hybrid subtracts the transmitted signal from the input signal, enabling the use of simple 100BASE-TX compatible magnetics.

3.5.7.8.3.2 Automatic Gain Control (AGC)

AGC normalizes the amplitude of the received signal, adjusting for the attenuation produced by the cable.

3.5.7.8.3.3 Timing Recovery

This function re-generates a receive clock from the incoming data stream which is used to sample the data. On the slave side of the link, this clock is also used to drive the transmitter.

3.5.7.8.3.4 Analog-to-Digital Converter (ADC)

The ADC function converts the incoming data stream from an analog waveform to digitized samples for processing by the DSP core.

3.5.7.8.3.5 Digital Signal Processor (DSP)

DSP provides per-channel adaptive filtering, which eliminates various signal impairments including:

- Inter-symbol interference (equalization)
- Echo caused by impedance mismatch of the cable
- Near-end crosstalk (NEXT) between adjacent channels (A, B, C, D)
- Far-end crosstalk (FEXT)
- Propagation delay variations between channels of up to 120 ns
- Extraneous tones that have been coupled into the receive path

The adaptive filter coefficients are initially set during the training phase. They are continuously adjusted (adaptive equalization) during operation through the decision-feedback loop.

3.5.7.8.3.6 De scrambler

The de scrambler identifies each channel by its characteristic signature, removing the signature and re-routing the channel internally. In this way, the receiver can correct for channel swaps and polarity reversals. The de scrambler uses the same base 33-bit LFSR used by the transmitter on the other side of the link.

The de scrambler automatically loads the seed value from the incoming stream of scrambled idle symbols. The de scrambler requires approximately 15 µs to lock, normally accomplished during the training phase.

3.5.7.8.3.7 Viterbi Decoder/Decision Feedback Equalizer (DFE)

The Viterbi decoder generates clean 4DPAM5 symbols from the output of the DSP. The decoder includes a Trellis encoder identical to the one used by the transmitter. The Viterbi decoder simultaneously looks at the received data over several baud periods. For each baud period, it predicts whether the symbol received should be even or odd, and compares that to the actual symbol received. The 4DPAM5 code is organized in such a way that a single level error on any channel changes an even code to an odd one and vice versa. In this way, the Viterbi decoder can detect single-level coding errors, effectively improving the signal-to-noise (SNR) ratio by a factor of 6 dB. When an error occurs, this information is quickly fed back into the equalizer to prevent future errors.

3.5.7.8.3.8 4DPAM5 Decoder

The 4DPAM5 decoder generates 8-byte data from the output of the Viterbi decoder.

3.5.7.8.3.9 100 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 de scrambles 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.

3.5.7.8.3.10 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.

3.5.7.8.3.11 Link Test

In 10 Mb/s mode, the PHY always transmits link pulses. If link test function is enabled, it monitors the connection for link pulses. Once it detects two to seven link pulses, data transmission are enabled and remain 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 the Port Configuration register bit (PHYREG.16.14) can disable the link test function.

3.5.7.8.3.12 10Base-T Link Failure Criteria and Override

Link failure occurs if link test is enabled and link pulses stop being received. If this condition occurs, the PHY returns to the auto-negotiation phase, if auto-negotiation is enabled. Setting the Port Configuration register bit (PHYREG.16.14) disables the link integrity test function, then the PHY transmits packets, regardless of link status.

3.5.7.8.3.13 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 bit PHYREG.16.10 = 1b.

3.5.7.8.3.14 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.

Automatic polarity correction can be disabled by setting bit PHYREG.27.5.

3.5.7.8.3.15 Dribble Bits

The PHY handles dribble bits for all of its modes. If between one and four dribble bits are received, the nibble is passed across the interface. The data passed across is padded with 1's if necessary. If between five and 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.

3.5.7.8.3.16 PHY Address

The PHY address for MDIO accesses is 00001b.

3.5.8 Media Auto Sense

The 82576 provides a significant amount of flexibility in pairing a LAN device with a particular type of media (such as 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/SGMII interface independently. The link configuration specified for each LAN device can be specified in the *LINK_MODE* field of the Extended Device Control (CTRL_EXT) register and initialized from the EEPROM Initialization Control Word 3 associated with each LAN device.

In some applications, software might need to be aware of the presence of a link on the media not currently active. In order to supply such an indication, any of the 82576 ports can set the *AUTOSENSE_EN* bit in the CONNSW register (address 0x00034) in order to enable sensing of the non active media activity.

Note: When in SerDes/SGMII detect mode, software should define which indication is used to detect the energy change on the SerDes/SGMII media. It can be either the external signal detect pin or the internal signal detect. This is done using the *CONNSW.ENRGSRC* bit. The signal detect pin is normally used when connecting in SerDes mode to optical media where the receive LED provide such an indication.

Software can then enable the OMED interrupt in ICR in order to get an indication on any detection of energy in the non active media.

Note: The auto-sense capability can be used in either port independent of the usage of the other port.

The following sections describes the procedures that should be followed in order to enable the auto-sense mode

3.5.8.1 Auto sense setup

3.5.8.1.1 SerDes/SGMII Detect Mode (PHY is active)

1. Set *CONNSW.ENRGSRC* to determine the sources for the signal detect indication (1b = external SIG_DET, 0b = internal SerDes electrical idle). The default of this bit is set by EEPROM.
2. Set *CONNSW.AUTOSENSE_EN*.
3. When link is detected on the SerDes /SGMII media, the 82576 sets the interrupt bit *OMED* in ICR and if enabled, issues an interrupt. The *CONNSW.AUTOSENSE_EN* bit is cleared .

3.5.8.1.2 PHY Detect Mode (SerDes/SGMII is active)

1. Set *CONNSW.AUTOSENSE_CONF* = 1b.
2. Reset the PHY as described in Section 4.2.
3. Place the PHY into link-disconnect mode by setting PHY_REG 25.5 using the MDIC register.
4. Set *CONNSW.AUTOSENSE_EN* = 1b and then clear *CONNSW.AUTOSENSE_CONF*.
5. When signal is detected on the PHY media, the 82576 sets the interrupt bit *OMED* in ICR and if enabled, issues an interrupt.
6. The 82576 puts the PHY in power down mode.

According to the result of the interrupt, software can then decide to switch to the other media.

3.5.8.2 Switching between medias.

The 82576's link mode is controlled by the Extended Device Control register; CTRL_EXT (0x00018) bits 23:22. The default value for the *LINK_MODE* setting is directly mapped from the EEPROM's initialization Control Word 3 (bits 1:0). Software can modify the *LINK_MODE* indication by writing the corresponding value into this register.

Note: Before dynamically switching between medias, the software should ensure that the current mode of operation is not in the process of transmitting or receiving data. This is achieved by disabling the transmitter and receiver, waiting until the 82576 is in an idle state, and then beginning the process for changing the link mode.

The mode switch in this method is only valid until the next hardware reset of the 82576. After a hardware reset, the link mode is restored to the default setting by the EEPROM. To get a permanent change of the link mode, the default in the EEPROM should be changed.

The following procedures need to be followed to actually switch between the two modes.

3.5.8.2.1 Transition to SerDes/SGMII mode

1. Disable the receiver by clearing *RCTL.RXEN*.
2. Disable the transmitter by clearing *TCTL.EN*.
3. Verify the 82576 has stopped processing outstanding cycles and is idle.
4. Modify LINK mode to SerDes or SGMII by setting *CTRL_EXT.LINK_MODE* to 11b or 10b, respectively.
5. Set up the link as described in Section 4.5.7.3 or Section 4.5.7.4.
6. Set up Tx and Rx queues and enable Tx and Rx processes.

3.5.8.2.2 Transition to internal PHY mode

1. Disable the receiver by clearing *RCTL.RXEN*.
2. Disable the transmitter by clearing *TCTL.EN*.
3. Verify the 82576 has stopped processing outstanding cycles and is idle.
4. Modify LINK mode to PHY mode by setting *CTRL_EXT.LINK_MODE* to 00b.
5. Set link-up indication by setting *CTRL.SLU*.
6. Reset the PHY as described in Section 4.2.
7. Set up the link as described in Section 4.5.7.4.
8. Set up the Tx and Rx qeues and enable the Tx and Rx processes.

§ §

4. Initialization

4.1 Power Up

4.1.1 Power-Up Sequence

Figure 4-1 shows the 82576 power-up sequence from power ramp up and until the device is ready to accept host commands.

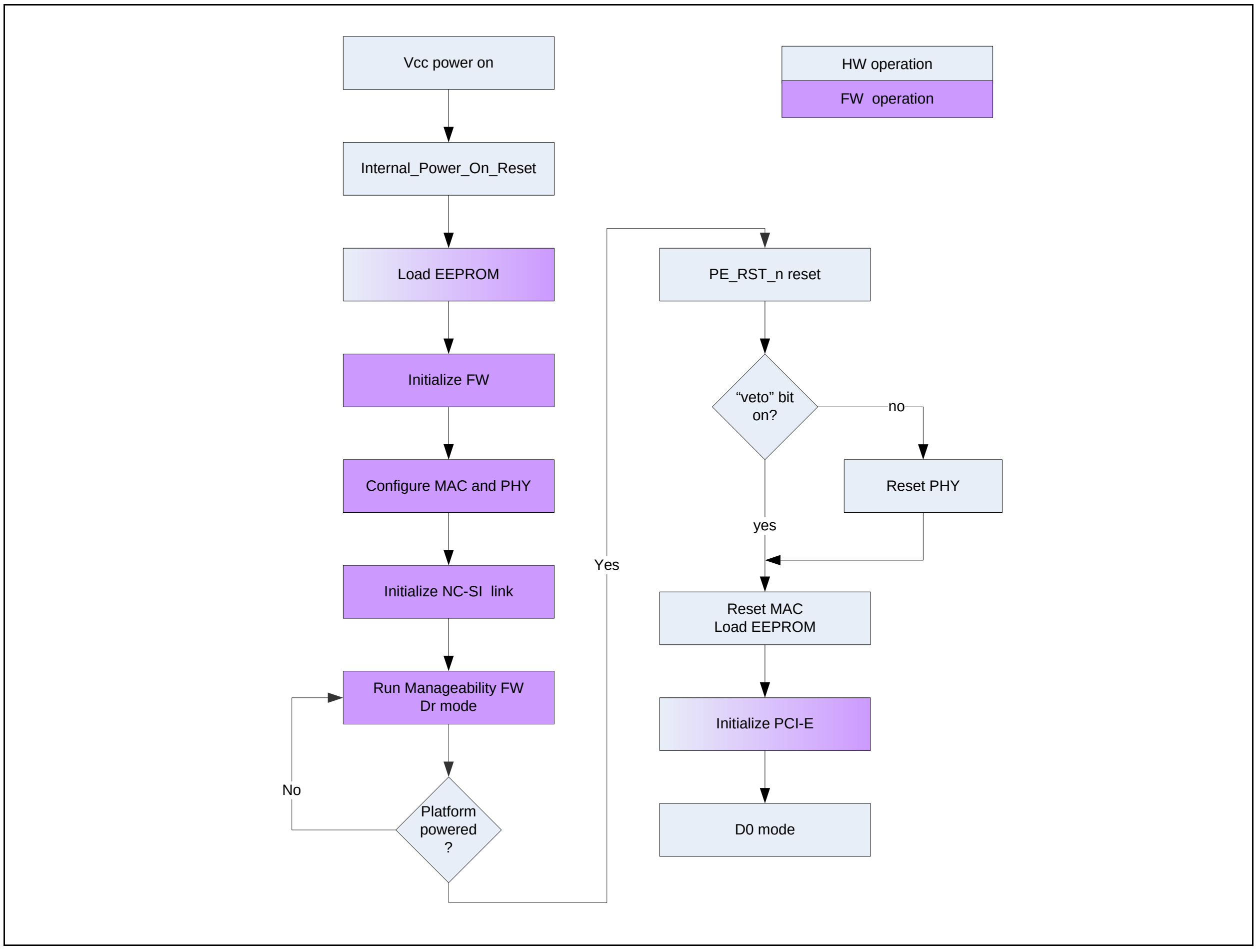


Figure 4-1. 82576 Power-Up - General Flow

Note: The keep_PHY_link_up bit (*Veto* bit) can be set by firmware when the BMC is running IDER or SoL. Its purpose is to prevent interruption of these processes when power is being turned on.

4.1.2 Power-Up Timing Diagram

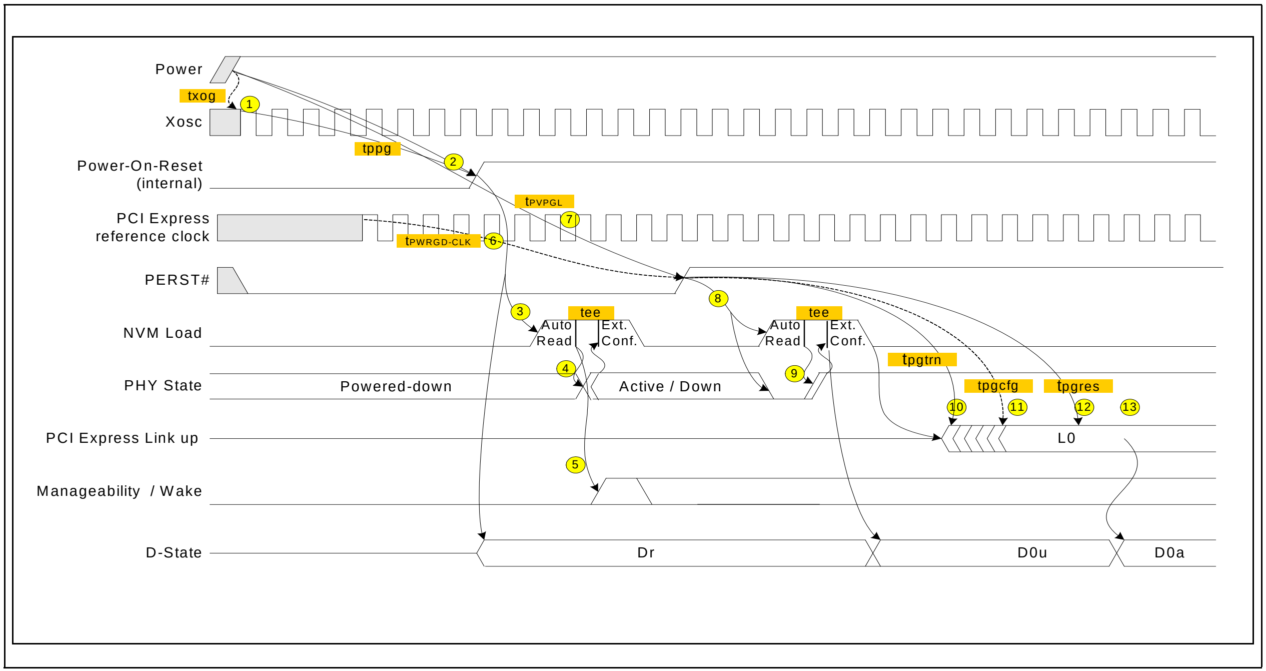


Figure 4-2. Power-Up Timing Diagram

Table 4-1. Notes to Power-Up Timing Diagram

Note	
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 .
4	After reading the NVM, PHY might exit power down mode.
5	APM Wakeup and/or manageability might be enabled based on NVM contents.
6	The PCIe reference clock is valid $t_{PE\_RST-CLK}$ before the de-assertion of PE_RST# (according to PCIe spec).
7	PE_RST# is de-asserted t_{PVPGL} after power is stable (according to PCIe spec).
8	De-assertion of PE_RST# causes the NVM to be re-read, asserts PHY power-down (except if veto bit also known as keep_PHY_link_up bit is set), and disables Wake Up.
9	After reading the NVM, PHY exits power-down mode.
10	Link training starts after t_{pgtrn} from PE_RST# de-assertion.
11	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST# de-assertion.
12	A first PCI configuration response can be sent after tpgres from PE_RST# de-assertion
13	Writing a 1 to the *Memory Access Enable* bit in the PCI *Command Register* transitions the device from D0u to D0 state.

4.1.2.1 Timing Requirements

the 82576 requires the following start-up and power state transitions.

Table 4-2. Power-Up Timing Requirements

Parameter	Description	Min.	Max.	Notes
t_{xog}	Base 25 clock stable from power stable		10msec	
$t_{PWRGD\text{-}CLK}$	PCIe* clock valid to PCIe* power good	100µs	-	According to PCIe* spec
t_{PVPGL}	Power rails stable to PCIe* PCIe Reset inactive	100ms	-	According to PCIe* spec
t_{pgcfg}	External PCIe Reset signal to first configuration cycle.	100ms		According to PCIe* spec

4.1.2.2 Timing Guarantees

the 82576 guarantees the following start-up and power state transition related timing parameters.

Table 4-3. Power-Up Timing Guarantees

Parameter	Description	Min.	Max.	Notes
t_{xog}	Xosc stable from power stable		10msec	
t_{ppg}	Internal power good delay from valid power rail		35msec	Use internal counter for external devices stabilization
t_{ee}	EEPROM read duration		20msec	Actual time depends on the EEPROM content
topll	PCIe* PCIe Reset to start of link training		10msec	
tpcipll	PCIe Reset to first configuration cycle		5msec	
tpgtrn	PCIe* PCIe Reset to start of link training		20msec	According to PCIe* spec
tpgres	PCIe Reset to first configuration cycle	100msec		According to PCIe* spec

4.2 Reset Operation

4.2.1 Reset Sources

The 82576 reset sources are described below:

4.2.1.1 Internal_Power_On_Reset

The 82576 has an internal mechanism for sensing the power pins. Once the power is up and stable the 82576 creates an internal reset, this reset acts as a master reset of the entire chip. It is level sensitive, and while it is zero holds all of the registers in reset. Internal_Power_On_Reset is interpreted to be an indication that device power supplies are all stable. Internal_Power_On_Reset changes state during system power-up.

4.2.1.2 PE_RST_N

The assertion of PE_RST_N 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 are reset on the rising edge of PE_RST_N. The only exception is the GIO unit, which is kept in reset while PE_RST_N is de-asserted (level).

4.2.1.3 In-Band PCIe Reset

The 82576 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.

4.2.1.4 D3hot α D0 Transition

This is also known as ACPI Reset. The 82576 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 transitions from D3hot to D0.

4.2.1.5 Function Level Reset (FLR)

The FLR bit is required for the PF and per VF (Virtual Function). Setting of this bit for a VF resets only the part of the logic dedicated to the specific VF and does not influence the shared part of the port. Setting the PF FLR bit resets the entire function.

4.2.1.5.1 PF (Physical Function) FLR or FLR in non-IOV Mode

An FLR reset to a function is equivalent to a D0 ⇨ D3 ⇨ D0 transition with the exception that this reset does not require driver intervention in order to stop the master transactions of this function. In an IOV enabled system, this reset resets all the VFs attached to the PF. The EEPROM is partially reloaded after an FLR reset.

The words read from EEPROM at FLR are the same read a full software reset.

4.2.1.5.2 VF (Virtual Function) FLR (Function Level Reset)

An FLR reset to a VF function resets all the queues, interrupts, and statistics registers attached to this VF. It also resets the PCIe R/W configuration bits allocated to this function. It also disables Tx & Rx flow for the queues allocated to this VF. All pending read requests are dropped and PCIe read completions to this function might be completed as unsupported requests.

4.2.1.5.3 IOV (IO Virtualization) Disable

Clearing of the IOV enable bit in the IOV structure is equivalent to a VFLR to all the active VFs in the PF.

4.2.1.6 Software Reset

4.2.1.6.1 Full Software Reset

Device Reset, RST, can be used to globally reset the entire component. This reset is provided primarily as a last-ditch software mechanism to recover from an indeterminate or suspected hung hardware state. Most registers (receive, transmit, interrupt, statistics, etc.), and state machines are set to their power-on reset values, approximating the state following a power-on or PCI reset. However, PCIe configuration registers are not reset, thereby leaving the device mapped into system memory space and accessible by a driver. One internal configuration register, the Packet Buffer Allocation (PBA) register, also retains its value through a global reset.

Note: To ensure that global device reset was fully completed and that the 82576 responds to subsequent accesses, wait approximately 1 millisecond after setting before attempting to check if the bit was cleared, or to access (read or write) any other device register.

Software can reset the 82576 by writing the Device Reset bit of the Device Control Register (CTRL.RST). The 82576 re-reads part of the per-function EEPROM fields after a software reset. Bits that are normally read from the EEPROM are reset to their default hardware values.

Fields controlled by the LED, SDP & Init3 words of the EEPROM are not reset and not re-read after a software reset.

Note: This reset is per function and resets only the function that received the software reset. PCI Configuration space (configuration and mapping) of the device is unaffected. Prior to issuing software reset the Driver needs to operate the master disable algorithm as defined in Section 5.2.3.2.

4.2.1.6.2 Physical Function (PF) Software Reset

A software reset by the PF in IOV mode has the same consequences as a software reset in a non-IOV mode. The procedure for PF software reset is as follows:

- The PF driver disables master accesses by the device through the Master Disable mechanism (see Section 5.2.3.2). Master Disable affects all VFs traffic.
- Execute the procedure described in Section 4.5.11.2.3 to synchronize between the PF and VFs.

VFs are expected to timeout and check on the *VFMailbox.RSTD* bit in order to identify a PF software reset event. The *VFMailbox.RSTD* bits are cleared on read.

4.2.1.6.3 VF Software Reset

A software reset applied to a VF is equivalent to an FLR reset to this VF with the exception that the PCIe configuration bits allocated to this function are not reset. This can be activated by setting the VFCTRL.RST bit.

4.2.1.7 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's Management Control word is 1. In pass through mode it is generated when receiving a ForceTCO SMB command with bit 1 or bit 7 set.

4.2.1.8 Firmware Reset

This reset is activated by writing a 1 to the FWR bit in the HOST Interface Control Register (HICR) in CSR address 0x8F00.

4.2.1.9 EEPROM Reset

Writing a 1 to the EEPROM Reset bit of the Extended Device Control Register (CTRL_EXT.EE_RST) causes the 82576 to re-read the per-function configuration from the EEPROM, setting the appropriate bits in the registers loaded by the EEPROM.

4.2.1.10 PHY Reset

Software can write a 1 to the PHY Reset bit of the Device Control Register (CTRL.PHY_RST) to reset the internal PHY. The PHY is internally configured after a PHY reset.

Note: The PHY should not be reset using PHYREG 0 bit 15, as in this case the internal configuration process is bypassed and there is no guarantee the PHY will operate correctly.

As the PHY may be accessed by the internal firmware and the driver software, the driver software should coordinate any PHY reset with the firmware using the following procedure:

1. Check that MANC.BLK_Phy_Rst_On_IDE (offset 0x5820 bit 18) is cleared. If it is set, the BMC requires a stable link and thus the PHY should not be reset at this stage. The driver may skip the PHY reset if not mandatory or wait for MANC.BLK_Phy_Rst_On_IDE to clear. See Section 4.2.3 for more details.
2. Take ownership of the relevant PHY using the following flow:

   a. Get ownership of the software/software semaphore SWSM.SMBI (offset 0x5B50 bit 0).
      - Read the SWSM register.
      - If SWSM.SMBI is read as zero, the semaphore was taken.
      - Otherwise, go back to step a.
      - This step assure that other software will not access the shared resources register (SW_FW_SYNC).
   b. Get ownership of the software/firmware semaphore SWSM.SWESMBI (offset 0x5B50 bit 1):
      - Set the SWSM.SWESMBI bit.
      - Read SWSM.
      - If SWSM.SWESMBI was successfully set - the semaphore was acquired - otherwise, go back to step a.
      - This step assure that the internal firmware will not access the shared resources register (SW_FW_SYNC).
   c. Software reads the Software-Firmware Synchronization Register (SW_FW_SYNC) and checks both bits in the pair of bits that control the PHY it wishes to own.
      - If both bits are cleared (both firmware and other software does not own the PHY), software sets the software bit in the pair of bits that control the resource it wishes to own.
      - If one of the bits is set (firmware or other software owns the PHY), software tries again later.
   d. Release ownership of the software/firmware semaphore by clearing the SWSM.SWESMBI bit.
3. Drive PHY reset bit in CTRL bit 31.
4. Wait 100 µs.
5. Release PHY reset in CTRL bit 31.
6. Release ownership of the relevant PHY to the FW using the following flow:
   e. Get ownership of the software/firmware semaphore SWSM.SWESMBI (offset 0x5B50 bit 1):
      - Set the SWSM.SWESMBI bit.
      - Read SWSM.
      - If SWSM.SWESMBI was successfully set - the semaphore was acquired - otherwise, go back to step a.
      - Clear the bit in SW_FW_SYNC that control the software ownership of the resource to indicate this resource is free.
      - Release ownership of the software/firmware semaphore by clearing the SWSM.SWESMBI bit.
7. Wait for the relevant CFG_DONE bit (EEMNGCTL.CFG_DONE0 - offset 0x1010 bit 18 or EEMNGCTL.CFG_DONE1 - offset 0x1010 bit 19).
8. Take ownership of the relevant PHY using the following flow:
   a. Get ownership of the software/firmware semaphore SWSM.SWESMBI (offset 0x5B50 bit 1):
      - Set the SWSM.SWESMBI bit.
      - Read SWSM.
      - If SWSM.SWESMBI was successfully set - the semaphore was acquired - otherwise, go back to step a.
      - This step assure that the internal firmware will not access the shared resources register (SW_FW_SYNC).
   b. Software reads the Software-Firmware Synchronization Register (SW_FW_SYNC) and checks both bits in the pair of bits that control the PHY it wishes to own.

- If both bits are cleared (both firmware and other software does not own the PHY), software sets the software bit in the pair of bits that control the resource it wishes to own.
- If one of the bits is set (firmware or other software owns the PHY), software tries again later.

c. Release ownership of the software/software semaphore and the software/firmware semaphore by clearing SWSM.SMBI and SWSM.SWESMBI bits.

9. Configure the PHY.
10. Release ownership of the relevant PHY using the flow described in Section 4.6.2.

4.2.2 Reset Effects

The resets affect the following registers and logic:

Table 4-4. 82576 Reset Effects - Common Resets

Reset Activation	Internal_Power _On_Reset	PE_ RST_N	In-Band PCIe Reset	FW Reset	Notes
LTSSM (PCIe back to detect/ polling)	X	X	X		
PCIe Link data path	X	X	X		
Read EEPROM (Per Function)					
Read EEPROM (Complete Load)	X	X	X		
PCI Configuration Registers- non sticky	X	X	X		3.
PCI Configuration Registers - sticky	X	X	X		4.
PCIe local registers	X	X	X		5.
Data path	X	X	X		
On-die memories	X	X	X		15.
MAC, PCS, Auto Negotiation, IPsec	X	X	X		
Virtual function queue enable	X	X	X		
Virtual function interrupt & statistics registers	X	X	X		13.
Wake Up (PM) Context	X	1			b
Wake Up Control Register	X				7.
Wake Up Status Registers	X				8.
Rule Checker Tables	X				
Manageability Control Registers	X				9.
MMS Unit	X			X	
Wake-Up Management Registers	X	X	X		3.,10.
Memory Configuration Registers	X	X	X		3.
EEPROM and flash request	X				16.
PHY/SERDES PHY	X	X	X		2.
Strapping Pins	X	X	X		

Table 4-5. 82576 Reset Effects - Per Function Resets

Reset Activation	D3hot –> D0	FLR	Full SW Reset	Force TCO	EE Reset	PHY Reset	Notes
Read EEPROM (Per Function)	X	X	X	X	X		
PCI Configuration Registers RO							3.
PCI Configuration Registers MSI-X	X	X					6.
PCI Configuration Registers RW							
PCIe local registers							5.
Data path	X	X	X	X			
On-die memories	X	X	X	X			15.
MAC, PCS, Auto Negotiation, IPsec	X	X	X	X			
Wake Up (PM) Context							b
Wake Up Control Register							7.
Wake Up Status Registers							8.
Rule Checker Tables							
Manageability Control Registers							9.
Virtual function queue enable	X	X	X	X			13.
Virtual function interrupt & statistics registers	X	X	X				13.
Wake-Up Management Registers	X	X	X	X			3.,10.
Memory Configuration Registers	X	X	X	X			3.
EEPROM and flash request	X	X					16.
PHY/SERDES PHY	X	X		X		X	2.
Strapping Pins							

Table 4-6. 82576 Reset Effects -Virtual Function Resets

Reset Activation	VFLR[17.]	Software Reset	Notes
Interrupt registers	X	X	13.
Queue disable	X	X	
VF specific PCIe configuration space	X		12.
Data path			

Notes:

1. If AUX_POWER = 0b the Wakeup Context is reset (PME_Status and PME_En bits should be 0b at reset if the 82576 does not support PME from D3cold).
2. The MMS unit must configure the PHY after any PHY reset.

3. The following register fields do not follow the general rules above:
   a. "CTRL.SDP0_IODIR, CTRL.SDP1_IODIR, CTRL_EXT.SDP2_IODIR, CTRL_EXT.SDP3_IODIR, CONNSW.ENRGSRC field, CTRL_EXT.SFP_Enable, CTRL_EXT.LINK_MODE, CTRL_EXT.EXT_VLAN and LED configuration registers are reset on Internal_Power_On_Reset only. Any EEPROM read resets these fields to the values in the EEPROM.
   b. The Aux Power Detected bit in the PCIe Device Status register is reset on Internal_Power_On_Reset and GIO Power Good only.
   c. The bits mentioned in the next note.
4. The following registers are part of this group:
   a. VPD registers
   b. Max payload size field in PCIe Capability Control register (offset 0xA8).
   c. Active State Link PM Control field, Common Clock Configuration field and Extended Synch field in PCIe Capability Link Control register (Offset 0xB0).
   d. ARI enable bit in IOV capability Command register (offset 0x168).
   e. Read Completion Boundary in the PCIe Link Control register (Offset 0xB0).
5. The following registers are part of this group:
   a. SWSM
   b. GCR (only part of the bits - see register description for details)
   c. FUNCTAG
   d. GSCL_1/2/3/4
   e. GSCN_0/1/2/3
   f. SW_FW_SYNC - only part of the bits - see register description for details.
6. The following registers are part of this group:
   a. MSIX control register, MSIX PBA and MSIX per vector mask.
   b. The Wake Up Context is defined in the PCI Bus Power Management Interface Specification (Sticky bits). It includes:
   — PME_En bit of the Power Management Control/Status Register (PMCSR).
   — PME_Status bit of the Power Management Control/Status Register (PMCSR).
   — Aux_En in the PCIe registers
   — The device Requester ID (since it is required for the PM_PME TLP).

   The shadow copies of these bits in the Wakeup Control Register are treated identically.
7. Refers to bits in the Wake Up Control Register that are not part of the Wake-Up Context (the PME_En and PME_Status bits).
8. The Wake Up Status Registers include the following:
   a. Wake Up Status Register
   b. Wake Up Packet Length.
   c. Wake Up Packet Memory.
9. The manageability control registers refer to the following registers:
   a. MANC 0x5820
   b. MFUTP01-7 0x5030 - 0x504C
   c. MFVAL 0x05824
   d. MANC2H 0x5860
   e. MAVTV1-7 0x5010 - 0x502C

   f. MDEF0-7 0x5890 - 0x58AC
   g. MDEF_EXT 0x5930 - 0x594C
   h. METF 0x5060 - 0x506C
   i. MIPAF0-15 0x58B0 - 0x58EC
   j. MMAH/MMAL0-3 0x5910 - 0x592C
   k. FWSM

10. The Wake-up Management Registers include the following:
    a. Wake Up Filter Control
    b. IP Address Valid
    c. IPv4 Address Table
    d. IPv6 Address Table
    e. Flexible Filter Length Table
    f. Flexible Filter Mask Table
11. The Other Configuration Registers includes:
    a. General Registers
    b. Interrupt Registers
    c. Receive Registers
    d. Transmit Registers
    e. Statistics Registers
    f. Diagnostic Registers

Of these registers, MTA[n], VFTA[n], WUPM[n], FFMT[n], FFVT[n], TDBAH/TDBAL, and RDBAH/RDVAL registers have no default value. If the functions associated with the registers are enabled they must be programmed by software. Once programmed, their value is preserved through all resets as long as power is applied to the 82576.

Note: In situations where the device is reset using the software reset CTRL.RST, the TX data lines is forced to all zeros. This causes a substantial number of symbol errors to be detected by the link partner. In TBI mode, if the duration is long enough, the link partner might restart the Auto-Negotiation process by sending “break-link” (/C/ codes with the configuration register value set to all zeros).

12. These registers includes
    a. MSI/MSI-X enable bits
    b. BME
    c. Error indications
13. These registers includes
    a. VTEICS
    b. VTEIMS
    c. VTEIAC
    d. VTEIAM
    e. VTEITR 0-2
    f. VTIVAR0
    g. VTIVAR_MISC
    h. PBACL

    i. VFMailbox

14. These registers includes
    a. RXDCTL.Enable
    b. Adequate bit in VFTE & VFRE.
15. The contents of the following memories are cleared to support the requirements of PCIe FLR:
    a. The Tx packet buffers
    b. The Rx packet buffers
    c. IPsec Tx SA tables
    d. IPsec Rx SA tables
16. Includes EEC.REQ, EEC.GNT, FLA.REQ and FLA.GNT fields.
17. A VFLR do not reset the configuration of the VF, only disables the interrupts and the queues.

4.2.3 PHY Behavior During a Manageability Session

During some manageability sessions (e.g. an IDER or SoL session as initiated by an external BMC), the platform is reset so that it boots from a remote media. This reset must not cause the Ethernet link to drop since the manageability session is lost. Also, the Ethernet link should be kept on continuously during the session for the same reasons. The 82576 therefore limits the cases in which the internal PHY would restart the link, by masking two types of events from the internal PHY:

- PE_RST# and PCIe resets (in-band and link drop) do not reset the PHY during such a manageability session
- The PHY does not change link speed as a result of a change in power management state, to avoid link loss. For example, the transition to D3hot state is not propagated to the PHY.
    - Note however that if main power is removed, the PHY is allowed to react to the change in power state (i.e., the PHY might respond in link speed change). The motivation for this exception is to reduce power when operating on auxiliary power by reducing link speed.

The capability described in this section is disabled by default on LAN Power Good reset. The Keep_PHY_Link_Up_En bit in the EEPROM must be set to '1' to enable it. Once enabled, the feature is enabled until the next LAN Power Good (i.e., the 82576 does not revert to the hardware default value on PE_RST#, PCIe reset or any other reset but LAN Power Good).

When the keep_PHY_link_up bit (also known as “veto bit”) in the MANC Register is set, the following behaviors are disabled:

- The PHY is not reset on PE_RST# and PCIe resets (in-band and link drop). Other reset events are not affected - LAN Power Good reset, Device Disable, Force TCO, and PHY reset by software.
- The PHY does not change its power state. As a result link speed does not change.
- The 82576 does not initiate configuration of the PHY to avoid losing link.

The keep_PHY_link_up bit is set by the BMC through the Management Control command (See Section 10.3.2.1.5 for SMBus command and Section 10.3.1.12.1 for NC-SI command) on the sideband interface. It is cleared by the external BMC (again, through a command on the sideband interface) when the manageability session ends. Once the keep_PHY_link_up bit is cleared, the PHY updates its Dx state and acts accordingly (e.g. negotiates its speed).

The keep_PHY_link_up bit is also cleared on de-assertion of the MAIN_PWR_OK input pin. MAIN_PWR_OK must be de-asserted at least 1 msec before power drops below its 90% value. This allows enough time to respond before auxiliary power takes over.

The keep_PHY_link_up bit is a R/W bit and can be accessed by host software, but software is not expected to clear the bit. The bit is cleared in the following cases:

- On LAN Power Good
- When the BMC resets or initializes it
- On de-assertion of the MAIN_PWR_OK input pin. The BMC should set the bit again if it wishes to maintain speed on exit from Dr state.

4.3 Function Disable

4.3.1 General

For a LOM (Lan on Motherboard) design, it might be desirable for the system to provide BIOS-setup capability for selectively enabling or disabling LAN functions. It allows the end-user more control over system resource-management and avoid conflicts with add-in NIC solutions. The 82576 provides support for selectively enabling or disabling one or both LAN device(s) in the system.

4.3.2 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 82576 LAN-disable mechanism is implemented in order to be compatible with such a solution.

The 82576 provides two mechanisms to disable LAN ports:

- Two pins (LANx_DIS_N, one per LAN port) are sampled on reset to determine the LAN-enable configuration
- Port 1 might be disabled using EEPROM configuration.

Disabling a LAN port affects the PCI function it resides on. When function 0 is disabled (either LAN0 or LAN1), two different behaviors are possible:

- Dummy Function mode — In some system, it is required to keep all the functions at their respective location, even when other functions are disabled. In Dummy Function mode, if function #0 (either LAN0 or LAN1) is disabled, then it does not disappear from the PCIe configuration space. Rather, the function presents itself as a dummy function. The device ID and class code of this function changes to other values (dummy function Device ID 0x10A6, Class Code 0xFF0000). In addition, the function does not require any memory or I/O space, and does not require an interrupt line.
- Legacy mode — When function 0 is disabled (either LAN0 or LAN1), then the port residing on function 1 moves to reside on function 0. Function 1 disappears from the PCI configuration space.

Note: In some systems, the dummy function is not recognized by the enumeration process as a valid PCI function. In these systems, both ports will not be enumerated and it is recommended to work in legacy mode.

The disabled LAN port is still available for manageability purposes if it was disabled using the LAN_PCI_DIS bit of the SDP control word in the EEPROM or if it was disabled through the pin mechanism and the PHY_in_LAN_Disable bit in the SDP control word in the EEPROM is cleared. In this case, and if LPLU bit is set, the PHY will attempt to create a link at 10 mbps.

Note: Dummy Function mode should not be used if SR-IOV capability is exposed (since PF0 is required to support certain functionality). SR-IOV is enabled by the IOV enable bit in EEPROM word 0x25 (Section 6.2.22).

Mapping between function and LAN ports is summarized in the following tables.

Table 4-7. PCI Functions Mapping (Legacy Mode)

PCI Function #	LAN Function Select	Function 0	Function 1
Both LAN functions are enabled	0	LAN 0	LAN 1
	1	LAN 1	LAN 0
LAN 0 is disabled	x	LAN1	Disable
LAN 1 is disabled	x	LAN 0	Disable
Both LAN functions are disabled	Both PCI functions are disabled. Device is in low power mode.		

Table 4-8. PCI Functions Mapping (Dummy Function Mode)

PCI Function #	LAN Function Select	Function 0	Function 1
Both LAN functions are enabled	0	LAN 0	LAN 1
	1	LAN 1	LAN 0
LAN 0 is disabled	0	Dummy	LAN1
	1	LAN 1	Disable
LAN 1 is disabled	0	LAN 0	Disable
	1	Dummy	LAN 0
Both LAN functions are disabled	Both PCI functions are disabled. Device is in low power mode.		

The following EEPROM bits control Function Disable:

- The access of the host through a PCI function to LAN1 can be enabled or disabled according to the “LAN PCI Disable” bit in EEPROM word 0x10 (Section 6.2.8).
- The “LAN Disable Select” EEPROM field in word 0x10 indicates if port 1 is disabled (Section 6.2.8).
- The “LAN Function Select” bit in EEPROM word 0x21 defines the correspondence between LAN Port and PCI function (Section 6.2.20)
- The “Dummy Function Enable” bit in EEPROM word 0x1B enables the Dummy Function mode. Default value is disabled (Section 6.2.16).
- The “PHY_in_LAN_disable” bit in EEPROM words 0x10 and 0x20 controls the availability of the disabled function to manageability channel when disabled through the LAN0_Dis_N or LAN1_Dis_N pins (Section 6.2.8 and Section 6.2.9).

When a particular LAN is fully disabled, all internal clocks to that LAN are disabled, the device is held in reset, and the internal PHY for that LAN is powered-down. In both modes, 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.

4.3.3 Control Options

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

Table 4-9. PCI functions Disable Options

Table 4-10. Strapping for Control Options

Function	Control Options
LAN 0	Strapping Option + EEPROM word 0x20 bit 13 (full/PCI only disable in case of strap)
LAN 1	Strapping Option + EEPROM word 0x10 bit 13 (full/PCI only disable in case of strap)/ EEPROM Word 0x10 bit 11 (full disable) / EEPROM word 0x10 bit 10 (PCI only disable)

The 82576 strapping option for LAN Disable feature:

Table 4-11. Strapping for LAN Disable

Symbol	Ball #	Name and Function
LAN0_Dis_N	B13	This pin is a strapping option pin always active. This pin has an internal weak pull-up resistor. In case this pin is not connected or driven hi during init time, LAN 0 is enabled. In case this pin is driven low during init time, LAN 0 is disabled. This pin is also used for testing and scan. When used for testing or scan, the LAN disable functionality is not active.
LAN1_Dis_N	A15	This pin is a strapping option pin always active. This pin has an internal weak pull-up resistor. In case this pin is not connected or driven hi during init time, LAN 1 is enabled. In case this pin is driven low during init time, LAN 1 function is disabled. This pin is also used for testing and scan. When used for testing or scan, the LAN disable functionality is not active.

4.3.4 Event Flow for Enable/Disable Functions

This section describes the driving levels and event sequence for device functionality. Following a Power on Reset / Internal Power / PE_RST_N/ In-Band reset the LANx_DIS_N signals should be driven hi (or left open) for nominal operation. If any of the LAN functions are not required statically its associated Disable strapping pin can be tied statically to low.

Case A - BIOS Disable the LAN Function at boot time by using strapping:

1. Assume that following power up sequence LANx_DIS_N signals are driven high.
2. The PCIe is established following the GIO_PWR_GOOD.
3. BIOS recognize that a LAN function in the 82576 should be disabled.
4. The BIOS drive the LANx_DIS_N signal to the low level.
5. The BIOS should assert the PCIe reset, either in-band or via PE_RST_N.
6. As a result, the 82576 samples the LANx_DIS_N signals and disable the LAN function and issue an internal reset to this function.
7. BIOS might start with the Device enumeration procedure (the disabled LAN function is invisible or changed to dummy function).
8. Proceed with Nominal operation.
9. Re-enable could be done by driving the LANx_DIS_N signal high and then request the user to issue a warm boot that generate bus enumeration.

4.3.4.1 Multi-Function Advertisement

If one of the LAN devices is disabled, the 82576 no longer is a multi-function device. The 82576 normally reports a 0x80 in the PCI Configuration Header field Header Type, indicating multi-function capability. However, if a LAN is disabled, the 82576 reports a 0x0 in this field to signify single-function capability.

4.3.4.2 Legacy Interrupts Utilization

When both LAN devices are enabled, the 82576 can utilizes INTA# to INTC# interrupts for interrupt reporting. The EEPROM Initialization Control Word 3 (bits 12:11) associated with each LAN device controls which of these interrupts are used for each LAN device. The specific interrupt pin utilized is reported in the PCI Configuration Header Interrupt Pin field associated with each LAN device.

However, if only one LAN device is enabled, then the INTA# must be used for this LAN device, regardless of the EEPROM configuration. Under these circumstances, the Interrupt Pin field of the PCI Header always reports a value of 0x1, indicating INTA# usage.

4.3.4.3 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 whenever the Data_Select field has a value of 0x8 (0x8 = Common Power Value Select).

When only one LAN is enabled, the 82576 appears as a single-function device, the Common Power value, if selected, reports 0x0 (undefined value), as Common Power is undefined for a single-function device.

4.4 Device 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 the end-user more control over system resource-management; avoid conflicts with add-in NIC solutions, etc. The 82576 provides support for selectively enabling or disabling it.

Note: If the 82576 is configured to provide a 50MHz NC-SI clock (via the NC-SI Output Clock EEPROM bit), then the device should not be disabled.

Device Disable is initiated by assertion of the asynchronous DEV_OFF_N pin. The DEV_OFF_N pin should always be connected to enable correct device operation.

The EEPROM “Device OFF En” bit enables device disable mode (hardware default is that the mode is disabled).

While in device disable mode, the PCIe link is in L3 state.The PHY is in power down mode. Output buffers are tri-stated.

Assertion or de-assertion of PCIe PE_RST_N does not have any effect while the device is in device disable mode (i.e., the device stays in the respective mode as long as DEV_OFF_N is asserted). However, the device might momentarily exit the device disable mode from the time PCIe PE_RST_N is de-asserted again and until the EEPROM is read.

During power-up, the DEV_OFF_N pin is ignored until the EEPROM is read. From that point, the device might enter Device Disable if DEV_OFF_N is asserted.

Note: De-assertion of the DEV_OFF_N pin causes a fundamental reset to the 82576.

Note to system designer: The DEV_OFF_N pin should maintain its state during system reset and system sleep states. It should also insure the proper default value on system power-up. For example, one could use a GPIO pin that defaults to '1' (enable) and is on system suspend power (i.e., it maintains state in S0-S5 ACPI states).

4.4.1 BIOS Handling of Device Disable

1. Assume that following power up sequence the DEV_OFF_N signal is driven high (else it is already disabled).
2. The PCIe is established following the GIO_PWR_GOOD.
3. BIOS recognize that the whole Device should be disabled.
4. The BIOS drive the DEV_OFF_N signal to the low level.
5. As a result, the 82576 samples the DEV_OFF_N signal and enters the device disable mode.
6. The BIOS put the Link in the Electrical IDLE state (at the other end of the PCIe link) by clearing the LINK Disable bit in the Link Control Register.
7. BIOS might start with the Device enumeration procedure (all of the Device functions are invisible).
8. Proceed with Nominal operation.
9. Re-enable could be done by driving the DEV_OFF_N signal high followed later by bus enumeration.

4.5 Software Initialization and Diagnostics

4.5.1 Introduction

This chapter discusses general software notes for the 82576, especially initialization steps. This includes general hardware, power-up state, basic device configuration, initialization of transmit and receive operation, link configuration, software reset capability, statistics, and diagnostic hints.

4.5.2 Power Up State

When the 82576 powers up it reads the EEPROM. The EEPROM contains sufficient information to bring the link up and configure the 82576 for manageability and/or APM wakeup. However, software initialization is required for normal operation.

The power-up sequence, as well as transitions between power states, are described in section 4.1.1. The detailed timing is given in Section 5.5. The next section gives more details on configuration requirements.

4.5.3 Initialization Sequence

The following sequence of commands is typically issued to device by the software device driver in order to initialize the 82576 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.

4.5.4 Interrupts During Initialization

- Most drivers disable interrupts during initialization to prevent re-entering to the interrupt routine. Interrupts are disabled by writing to the IMC register. Note that the interrupts need to be disabled also after issuing a global reset, so a typical driver initialization flow is:

- Disable interrupts
- Issue a Global Reset
- Disable interrupts (again)
- ...

After the initialization is done, a typical driver enables the desired interrupts by writing to the IMS register.

4.5.5 Global Reset and General Configuration

Device initialization typically starts with a global reset that puts the device into a known state and enables the device driver to continue the initialization sequence.

Several values in the Device Control Register (CTRL) need to be set, upon power up, or after a device reset for normal operation.

- 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 PCS layer in SGMII/SerDes mode, or forced by software if the link is forced. Status information for speed is also readable in STATUS.
- The device defaults to LRST = 1 which disables Auto-Negotiation. A transition to 0 initiates the Auto-Negotiation function. LRST might be defined in the EEPROM.
- ILOS should normally be set to 0.

Set the packet buffer allocation for transmit and receive flows in the RXPBS, TXPBS & SWPBS registers. This should be done before RCTL.RXEN & TCTL.TXEN are set. An ordered disabling of all queues and of the Rx & Tx flows is required before any change in the packet buffer allocation is done.

4.5.6 Flow Control Setup

If flow control is enabled, program the FCRTL, FCRTH, FCTTV and FCRTV registers. In order to avoid packet losses, FCRTH should be set to a value equal to at least two max size packet below the receive buffer size. E.g. Assuming a packet buffer size of 32K and expected max size packet of 9.5K, the FCRTH value should be set to 32 - 2 * 9.5 = 14K i.e. RTH should be set to 0x380.

4.5.7 Link Setup Mechanisms and Control/Status Bit Summary

Note: The CTRL_EXT.LINK_MODE value should be set to the desired mode prior to the setting of the other fields in the link setup procedures.

4.5.7.1 PHY Initialization

Refer to the PHY documentation for the initialization and link setup steps. The device driver uses the MDIC register to initialize the PHY and setup the link. Section 3.5.4.3 describes the link setup for the internal copper PHY. Section 3.5.2.2 describes the usage of the MDIC register.

4.5.7.2 MAC/PHY Link Setup (CTRL_EXT.LINK_MODE = 00)

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).

4.5.7.2.1 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 1 by software to enable communications between MAC and PHY.

CTRL.RFCE Must be set by S/W after reading flow control resolution from PHY registers.

CTRL.TFCE Must be set by S/W after reading flow control resolution from PHY registers.

CTRL.SPEED Don't care; speed setting is 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 1).

STATUS.SPEED Reflects actual speed setting negotiated by the PHY and indicated to the MAC (SPD_IND).

4.5.7.2.2 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 1 by software to enable communications between MAC and PHY.

CTRL.RFCE Must be set by S/W after reading flow control resolution from PHY registers.

CTRL.TFCE Must be set by S/W after reading flow control resolution from PHY registers.

CTRL.SPEED Set by software based on reading PHY status register after PHY has 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 1).

STATUS.SPEED Reflects MAC forced speed setting written in CTRL.SPEED.

4.5.7.2.3 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 1 by software to enable communications between MAC and PHY. PHY must also be forced/configured to indicate positive link indication (LINK) to the MAC.

CTRL.RFCE Must be set by S/W to desired flow-control operation (must match flow-control settings of PHY).

CTRL.TFCE Must be set by S/W to desired flow-control operation (must match flow-control settings of PHY).

CTRL.SPEED Set by software to desired link speed (must match speed setting of PHY).

STATUS.FD Reflects the MAC duplex setting written by software to CTRL.FD.

STATUS.LU Reflects 1 (positive link indication LINK from PHY qualified with CTRL.SLU). Note that 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.

4.5.7.3 MAC/SERDES Link Setup (CTRL_EXT.LINK_MODE = 11b)

Link setup procedures using an external SERDES interface mode:

4.5.7.3.1 Hardware Auto-Negotiation Enabled (PCS_LCTL. AN ENABLE = 1b; CTRL.FRCSPD = 0b; CTRL.FRCDPLX = 0)

CTRL.FD	Ignored; duplex is set by priority resolution of PCS_ANDV and PCS_LPAB.
CTRL.SLU	Must be set to 1 by software to enable communications to the SerDes.
CTRL.RFCE	Set by Hardware according to auto negotiation resolution[1].
CTRL.TFCE	Set by Hardware according to auto negotiation resolution[1].
CTRL.SPEED	Ignored; speed always 1000Mb/s when using SGMII mode communications.
STATUS.FD	Reflects hardware-negotiated priority resolution.
STATUS.LU	Reflects PCS_LSTS.AN COMPLETE (Auto-Negotiation complete).
STATUS.SPEED	Reflects 1000Mb/s speed, reporting fixed value of (10)b.
PCS_LCTL.FSD	Must be zero.
PCS_LCTL.Force Flow Control	Must be zero[1].
PCS_LCTL.FSV	Must be set to 10b. Only 1000 Mb/s is supported in SerDes mode.
PCS_LCTL.FDV	Ignored; duplex is set by priority resolution of PCS_ANDV and PCS_LPAB.

4.5.7.3.2 Auto-Negotiation Skipped (PCS_LCTL. AN ENABLE = 0b;CTRL.FRCSPD = 1b; CTRL.FRCDPLX = 1)

CTRL.FD	Must be set to 1b. - only full duplex is supported in SerDes mode.
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	Must be set to 10b. Only 1000 Mb/s is supported in SerDes mode.
STATUS.FD	Reflects the value written by software to CTRL.FD.
STATUS.LU	Reflects whether the PCS detected comma symbols, qualified with CTRL.SLU (set to 1).
STATUS.SPEED	Reflects 1000Mb/s speed, reporting fixed value of (10)b.
PCS_LCTL.FSD	Must be set to 1 by software to enable communications to the SerDes.
PCS_LCTL.Force Flow Control	Must be set to 1.
PCS_LCTL.FSV	Must be set to 10b. Only 1000 Mb/s is supported in SerDes mode.
PCS_LCTL.FDV	Must be set to 1b - only full duplex is supported in SerDes mode.

4.5.7.4 MAC/SGMII Link Setup (CTRL_EXT.LINK_MODE = 10b)

Link setup procedures using an external SGMII interface mode:

[1] If PCS_LCTL.Force Flow Control is set, the auto negotiation result is not reflected in the CTRL.RFCE and CTRL.TFCE registers. In This case, the software must set these fields after reading flow control resolution from PCS registers.

4.5.7.4.1 Hardware Auto-Negotiation Enabled (PCS_LCTL. AN ENABLE = 1b, CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 0b)

CTRL.FD Ignored; duplex is set by priority resolution of PCS_ANDV and PCS_LPAB.

CTRL.SLU Must be set to 1 by software to enable communications to the SerDes.

CTRL.RFCE Must be set by software after reading flow control resolution from PCS registers.

CTRL.TFCE Must be set by software after reading flow control resolution from PCS registers.

CTRL.SPEED Ignored; speed setting is established from SGMII's internal indication to the MAC after SGMII has auto-negotiated a successful link-up.

STATUS.FD Reflects hardware-negotiated priority resolution.

STATUS.LU Reflects PCS_LSTS.Link OK

STATUS.SPEED Reflects actual speed setting negotiated by the SGMII and indicated to the MAC.

PCS_LCTL.Force Flow Control Ignored.

PCS_LCTL.FSD Should be set to zero.

PCS_LCTL.FSV Ignored; speed is set by priority resolution of PCS_ANDV and PCS_LPAB.

PCS_LCTL.FDV Ignored; duplex is set by priority resolution of PCS_ANDV and PCS_LPAB.

4.5.8 Initialization of Statistics

Statistics registers are hardware-initialized to values as detailed in each particular register's description. The initialization of these registers begins upon transition to 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 µsec. of this transition. Access to statistics registers prior to this interval might return indeterminate values.

All of the statistical counters are cleared on read and a typical device driver reads them (thus making them zero) as a part of the initialization sequence.

4.5.9 Receive Initialization

Program the Receive address register(s) per the station address. This can come from the EEPROM or from any other means (for example, on some machines, this comes from the system PROM not the EEPROM on the adapter card).

Set up the MTA (Multicast Table Array) per software. This means zeroing all entries initially and adding in entries as requested.

Program RCTL 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.

The following should be done once per receive queue needed:

- 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.
- Set the length register to the size of the descriptor ring.
- Program SRRCTL of the queue according to the size of the buffers and the required header handling.

- If header split or header replication is required for this queue, program the PSRTYPE register according to the required headers.
- Enable the queue by setting RXDCTL.ENABLE. In the case of queue zero, the enable bit is set by default - so the ring parameters should be set before RCTL.RXEN is set.
- Poll the RXDCTL register until the ENABLE bit is set. The tail should not be bumped before this bit was read as one.
- Program the direction of packets to this queue according to the mode select in MRQC. Packets directed to a disabled queue is dropped.

Note: The tail register of the queue (RDT[n]) should not be bumped until the queue is enabled.

4.5.9.1 Initialize the Receive Control Register

To properly receive packets the receiver should be enabled by setting RCTL.RXEN. 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.

4.5.9.2 Dynamic Enabling and Disabling of Receive Queues

Receive queues can be dynamically enabled or disabled given the following procedure is followed:

Enabling:

- Follow the per queue initialization described in the previous section.
- Note that if there are still packets in the packet buffer directed to this queue according to previous settings, they is received after the queue is re-enabled. In order to avoid this condition, the software might poll the PBRWAC register. Once a wrap around or an empty condition of the relevant packet buffer is detected the queue might be re-enabled.

Disabling:

- Disable the direction of packets to this queue.
- Disable the queue by clearing RXDCTL.ENABLE. The 82576 stops fetching and writing back descriptors from this queue immediately. The 82576 eventually completes the storage of one buffer allocated to this queue. Any further packet directed to this queue is dropped. If the currently processed packet is spread over more than one buffer, all subsequent buffers is not written.
- The 82576 clears RXDCTL.ENABLE only after all pending memory accesses to the descriptor ring or to the buffers are done. The driver should poll this bit before releasing the memory allocated to this queue.

The RX path might be disabled only after all Rx queues are disabled.

4.5.10 Transmit Initialization

Program the TCTL register according to the MAC behavior needed.

If work in half duplex mode is expected, program the TCTL_EXT.COLD field. For internal PHY mode the default value of 0x41 is OK. For SGMII mode, a value reflecting the 82576 and the PHY SGMII delays should be used. A suggested value for a typical PHY is 0x46 for 10 Mbps and 0x4C for 100 Mbps.

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 to the size of the descriptor ring.

- Program the TXDCTL register with the desired TX descriptor write back policy. Suggested values are:
  - — WTHRESH = 1b
  - — All other fields 0b.
- If needed, set the TDWBAL/TWDBAH to enable head write back
- Enable the queue using TXDCTL.ENABLE (queue zero is enabled by default).
- Poll the TXDCTL register until the ENABLE bit is set.

Note: The tail register of the queue (TDT[n]) should not be bumped until the queue is enabled.

Enable transmit path by setting TCTL.EN. This should be done only after all other settings are done.

4.5.10.1 Dynamic Queue Enabling and Disabling

Transmit queues can be dynamically enabled or disabled given the following procedure is followed:

Enabling:

- Follow the per queue initialization described in the previous section.

Disabling:

- Stop storing packets for transmission in this queue.
- Wait until the head of the queue (TDH) is equal to the tail (TDT), i.e. the queue is empty.
- Disable the queue by clearing TXDCTL.ENABLE.

The Tx path might be disabled only after all Tx queues are disabled.

4.5.11 Virtualization Initialization Flow

4.5.11.1 Next Generation VMDq Mode

4.5.11.1.1 Global Filtering and Offload Capabilities

- Select one of the Next Generation VMDq pooling methods - MAC/VLAN filtering for pool selection and RSS for the queue in pool selection. MRQC.Multiple Receive Queues Enable = 011b, 100b or 101b.
- In RSS mode, the RSS key (RSSRK) and redirection table (RETA) should be programmed. Note that the redirection table is common to all the pools and only indicates the queue inside the pool to use once the pool is chosen.
- Set the RPLOLR and RPLPSRTYPE registers to define the behavior of replicated packets.
- Configure VT_CTL.DEF_PL to define the default pool. If packets with no pools should be dropped, set VT_CTL.Dis_def_Pool field.
- Enable replication via VT_CTL.replication_en.
- Enable loopback via DTXSWC.Loopback.
- If needed, enable padding of small packets via the RCTL.PSP

4.5.11.1.2 Mirroring rules.

For each mirroring rule to be activated:

a. Set the type of traffic to be mirrored in the VMRCTL[n] register.

b. Set the mirror pool in the VMRCTL[n].MP

c. For pool mirroring, set the VMRVM[n] register with the pools to be mirrored.

d. For VLAN mirroring, set the VMVRLAN[n] with the indexes from the VLVF registers of the VLANs to be mirrored.

4.5.11.1.3 Per Pool Settings

As soon as a pool of queues is associated to a VM the software should set the following parameters:

1. Address filtering:
    a. The unicast MAC address of the VM by enabling the pool in the RAH/RAL registers.
    b. If all the MAC addresses are used, the unicast hash table (UTA) can be used. Pools servicing VMs whose address is in the hash table should be declared as so by setting the VMOLR.ROPE. Packets received according to this method didn't pass perfect filtering and are indicated as such.
    c. Enable the pool in all the RAH/RAL registers representing the multicast MAC addresses this VM belongs to.
    d. If all the MAC addresses are used, the multicast hash table (MTA) can be used. Pools servicing VMs using multicast addresses in the hash table should be declared as so by setting the VMOLR.ROMPE. Packets received according to this method didn't pass perfect filtering and are indicated as such.
    e. Define whether this VM should get all multicast/broadcast packets in the same VLAN via the *VMOLR.MPE* and *VMOLR.BAM fields*
    f. Enable the pool in each *VLVF* register representing a VLAN this VM belongs to.
    g. A VM might be set to receive it's own traffic in case the source and the destination are in the same pool via the *DTXSWC.LLE* field.
    h. Define whether the pool belongs to the default VLAN and should accept untagged packets via the *VMOLR.AUPE* field
2. Offloads
    a. Define whether VLAN header should be stripped from the packet. CRC is always stripped from the packet.
    b. Set which header split is required via the PSRTYPE register.
    c. Set whether larger than standard packet are allowed by the VM and what is the largest packet allowed (jumbo packets support) via VMOLR.RLPML & VMOLR.RLE.
    d. In RSS mode, define if the pool uses RSS via the VMOLR.RSSE bit.
3. Queues
    a. Enable Rx & Tx queues as described in Section 4.5.9 & Section 4.5.10
    b. For each Rx queue a drop/no drop flag can be set in SRRCTL.DROP_EN or via the QDE register, controlling the behavior in cases no receive buffers are available in the queue to receive packets. The usual behavior is to allow drops in order to avoid head of line blocking, unless a no-drop behavior is needed for some type of traffic (e.g. storage).

4.5.11.1.4 Security Features

4.5.11.1.4.1 Anti spoofing

For each pool, the driver may activate the MAC and VLAN anti spoof features via the relevant bit in *DTXSWC.MACAS* and *DTXSWC.VLANAS* respectively.

4.5.11.1.4.2 Storm control

The driver may set limits to the broadcast or multicast traffic it can receive.

1. It should set how many 64 bytes chunks of Broadcast and Multicast traffic are acceptable per interval via the *BSCTRH* and *MSCTRH* respectively.
2. It should then set the interval to be used via the *SCCRL.Interval* field and which action to take when the broadcast or multicast traffic crosses the programmed threshold via the *SCCRL.BDIPW, SCCRL.BDICW, SCCRL.MDIPW,* and *SCCRL.MDICW* fields.
3. The driver may be notified of storm control events through the *ICR.SCE* interrupt cause.

4.5.11.1.5 Allocation of Tx Bandwidth to VMs

4.5.11.1.5.1 Configuring Tx Bandwidth to VMs

Allocation of Tx Bandwidth to VMs feature is enabled or disabled via the programming of VMBACS and VMBAMMW registers. When enabled, bandwidth to VMs (i.e. to Tx Queues) is configured via writing into VMBASEL and VMBAC registers for each queue again.

The bandwidth configuring procedure is as follow -

1. Allocate non-null rates to VMs present in the system RVMi (i=0..up to 7), in Gb/s units, so that:

   RVM0 + RVM1 + ... + RVM7 = 0.5 Gb/s

   Assume also that for any different i,j:

   RVMi / RVMj < 10 and RVMj / RVMi < 10

2. Allocate rates to enabled queues RQi (i=0..up to 7), in Gb/s units, so that:

   $RQ_i = RQ_{i+8} = RVM_i / 2$

3. Compute rate factors RFQi (i=0..15) for all the enabled Tx queues, so that:

   RFQi = 1 Gb/s / RQi

4. Format the rate factors obtained in the previous step as decimal binary numbers, with 10-bits integral part *left* of the decimal point, and 14-bits decimal part *right* of it, and for i=0..15, set RTTDQSEL.TXDQ_IDX=i and then:
   a. Set RTTDVMRC.RF_INT = integral part of RFQi
   b. Set RTTDVMRC.RF_DEC = decimal part of RFQi
5. Compute VM_MMW_SIZE to the VM Rate-Scheduler as follow:

   VM_MMW_SIZE = 16 x MSS

   for avoiding saturation while full workload. Refer to Section 4.5.11.1.5.2.
6. Set VMBAMMW.MMW_SIZE = VM_MMW_SIZE

4.5.11.1.5.2 Link Speed Change Procedure

Whenever the link status or speed is changed, the 82576 operates the VM arbiters in a packet based round robin mode, and disables the VM rate-controllers. Software is responsible to re-enabling and re-configuring them accordingly to the new link speed. However, to avoid any race condition between hardware and software, the following procedure must be performed by the driver whenever a link speed/status change interrupt occurs:

1. Check the SPEED_CHG bit in VMBACS register was asserted by hardware.
2. Read the VMBA_SET bit in the VMBACS register.

3. If the bit is read as 1, it means the VM rate-controllers were not completely disabled by hardware (i.e. a race occurred between hardware and software). Software must therefore clear the RC_ENA bit in the VMBAC register for all the queues, or for at least the queue(s) for which it is still set.
4. Clear the SPEED_CHG bit in VMBACS register.

4.5.11.2 IOV Initialization

The initialization flow used to enable an IOV function can be found in chapter 2 of the PCI-Express Single Root I/O Virtualization and Sharing Specification.

4.5.11.2.1 PF Driver Initialization

The PF driver is responsible for the link setup and handling of all the filtering & off load capabilities for all the VFs as described in Section 4.5.11.1.1 and the security features as described in Section 4.5.11.1.4. It should also set the bandwidth allocation per transmit queue for each VF as described in Section 4.5.10 and Section 4.5.11.1.5.

Note: The link setup might include authentication process (802.1X or other).

In IOV mode, Next Generation VMDq + RSS mode is not available. RSS mode might be used, but this assumes all the VMs uses the same key, RSS hash algorithms and redirection table which is currently not POR of any VMM vendor.

After all the common parameters are set, the PF driver should set all the VFMailbox.RSTD bit by setting the CTRL.PFRSTD.

The PF might disable all the active VFs traffic via the VFTE & RFTE registers until the parameters of a VF are set as defined in Section 4.5.11.1.3, the VF can be enabled using the same registers.

4.5.11.2.1.1 VF Specific Reset Coordination

After the PF driver receives an indication of a VF FLR via the VFLRE register, it should enable the receive and transmit for the VF only once the device is programmed with the right parameters as defined in Section 4.5.11.1.3. The receive filtering is enabled using the VFRE register and the transmit filtering is enabled via the VFTE register.

Note: The filtering & off loads setup might be based on a central IT settings or on requests from the VF drivers.

The PF driver should assert the VF reset via the VCTRL register before configuration of the VF parameters.

4.5.11.2.2 VF Driver Initialization

Upon init, after the PF indicated that the global init was done via the VFMailbox.RSTD bit, the VF driver should communicate with the PF, either via the mailbox or other software mechanisms to assure that the right parameters of the VF are programmed as described in Section 4.5.11.1.3.

The mailbox mechanism is described in Section 7.9.2.9.1.

The PF should also setup the security measures as described in Section 4.5.11.1.4. In addition, the PF may also program whether the VF is allowed to control VLAN insertion or whether VLAN insertion is controlled by the PF via the relevant *VMVIR* register.

The PF driver might then send an acknowledge message with the actual setup done according to the VF request and the IT policy.

The VF driver should then setup the interrupts and the queues as described in Section 4.5.9 & Section 4.5.10.

4.5.11.2.3 Full Reset Coordination

A mechanism is provided to synchronize reset procedures between the Physical Function and the VFs. It is provided specifically for PF software reset but can be used in other reset cases as described below.

The procedure is as follows:

One of the following reset cases takes place:

- Internal_Power_On_Reset
- PCIe Reset (PERST and in-band)
- D3hot --> D0
- FLR
- Software reset by the PF

The 82576 sets the RSTI bits in all the VFMailbox registers. Once the reset completes, each VF might read its VFMailbox register to identify a reset in progress.

Once the PF completed configuring the device, it sets the CTRL_EXT.PFRSTD bit. As a result, the 82576 clears the RSTI bits in all the VFMailbox registers and sets the RSTD (Reset Done) bits are set in all the VFMailbox registers.

Until a RSTD condition is detected, the VFs should access only the VFMailbox register and should not attempt to activate the interrupt mechanism or the transmit and receive process.

4.5.11.2.4 IOV disable

After IOV is disabled, the PF can not immediately reuse the resources released by the VF. It should first wait 100 ms, to make sure all the pending request and completions of the defunct VFs are processed. After that, it should set the *IOVCTL.Use VF Queues* bit. Only then, the released queues may be reused by the PF.

4.5.11.2.5 VFRE / VFTE

This mechanism insures that a VF cannot transmit or receive before the Tx and Rx path have been initialized by the PF. It is required for VFLR reset and must also be used in case of VF software reset. It is optional for PF software reset as described above. The VFRE register contains a bit per VF. When the bit is cleared assignment of Rx packet for the VF's pool is disabled. When set, assignment of Rx packet for the VF's pool is enabled.

The VFTE register contains a bit per VF. When the bit is cleared, fetching of data for the VF's pool is disabled. When set, fetching of data for the VF's pool is enabled. Fetching of descriptors for the VF pool is maintained, up to the limit of the internal descriptor queues - regardless to VFTE settings.

Note: The VFRE and VFTE registers apply in all device modes (not just IOV). The default values for both registers are therefore ‘1’, enabling transmission and reception in non-IOV modes.

4.6 Access to shared resources

Part of the resources in the 82576 are shared between several software entities - namely the drivers of the two ports and the internal firmware. In order to avoid contentions, a driver that needs to access one of these resources should use the flow described in Section 4.6.1 in order to acquire ownership of this resource and use the flow described in Section 4.6.2 in order to relinquish ownership of this resource.

The shared resources are:

1. The EEPROM
2. Both PHYs
3. CSRs accessed by the internal firmware after the initialization process. Currently there are no such CSRs.
4. The flash.

Note: Any other software tool that access the the 82576 register set directly should also follow the flow described below.

4.6.1 Acquiring ownership over a shared resource

The following flow should be used to acquire a shared resource:

1. Get ownership of the software/software semaphore SWSM.SMBI (offset 0x5B50 bit 0).
    a. Read the SWSM register.
    b. If SWSM.SMBI is read as zero, the semaphore was taken.
    c. Otherwise, go back to step a.

    This step assure that other software will not access the shared resources register (SW_FW_SYNC).
2. Get ownership of the software/firmware semaphore SWSM.SWESMBI (offset 0x5B50 bit 1):
    a. Set the SWSM.SWESMBI bit.
    b. Read SWSM.
    c. If SWSM.SWESMBI was successfully set - the semaphore was acquired - otherwise, go back to step a.

    This step assure that the internal firmware will not access the shared resources register (SW_FW_SYNC).
3. Software reads the Software-Firmware Synchronization Register (SW_FW_SYNC) and checks both bits in the pair of bits that control the resource it wishes to own.
    a. If both bits are cleared (both firmware and other software does not own the resource), software sets the software bit in the pair of bits that control the resource it wishes to own.
    b. If one of the bits is set (firmware or other software owns the resource), software tries again later.
4. Release ownership of the software/software semaphore and the software/firmware semaphore by clearing SWSM.SMBI and SWSM.SWESMBI bits.
5. At this stage, the shared resources is owned by the driver and it may access it. The SWSM and SW_FW_SYNC registers can now be used to take ownership of another shared resources.

4.6.2 Releasing ownership over a shared resource

The following flow should be used to release a shared resource:

1. Get ownership of the software/software semaphore SWSM.SMBI (offset 0x5B50 bit 0).
    a. Read the SWSM register.
    b. If SWSM.SMBI is read as zero, the semaphore was taken.

   c. Otherwise, go back to step a.

   This step assure that other software will not access the shared resources register (SW_FW_SYNC).

2. Get ownership of the software/firmware semaphore SWSM.SWESMBI (offset 0x5B50 bit 1):
   a. Set the SWSM.SWESMBI bit.
   b. Read SWSM.
   c. If SWSM.SWESMBI was successfully set - the semaphore was acquired - otherwise, go back to step a.

   This step assure that the internal firmware will not access the shared resources register (SW_FW_SYNC).

3. Clear the bit in SW_FW_SYNC that control the software ownership of the resource to indicate this resource is free.
4. Release ownership of the software/software semaphore and the software/firmware semaphore by clearing SWSM.SMBI and SWSM.SWESMBI bits.
5. At this stage, the shared resources is released by the driver and it may not access it. The SWSM and SW_FW_SYNC registers can now be used to take ownership of another shared resources.

§ §

5. Power Management

This section describes how power management is implemented in the 82576. The 82576 supports the Advanced Configuration and Power Interface (ACPI) specification as well as Advanced Power Management (APM).

Note: Power management can be disabled via bits in the Initialization Control Word, which is loaded from the EEPROM during power-up reset. Even when disabled, the power management register set is still present. Power management support is required by the PCIe specification.

5.1 General Power State Information

5.1.1 PCI Device Power States

The ACPI specification defines component device power states (D-states) that enable the platform to establish and control power states for the 82576 ranging from fully on to fully off (drawing no power) and various in-between levels of power-saving states, annotated as D0-D3. Similarly, PCIe defines a series of link power states (L-states) that work specifically within the link layer between the 82576 and its upstream PCIe port (typically in the host chipset). For a given component D-state, only certain L-states are possible as follows.

- D0 (fully on): The 82576 is completely active and responsive during this D-state. The link can be in either L0 or a low-latency idle state referred to as L0s. Minimizing L0s exit latency is paramount for enabling frequent entry into L0s while facilitating performance needs via a fast exit. The next lower link power state, L1 state, can be achieved either by hardware-based ASPM or by requesting the link to enter L1 after the operating system places the downstream 82576 in a D1-D3 state.
- D1 and D2: These modes are not supported by the 82576.
- D3 (off): Primary power can be fully removed from the 82576 [D3cold (Dr)], or not removed from the 82576 (D3hot). D3cold maps to L2 if auxiliary power is supported on the 82576 with wake-capable logic, or to L3 if no power is delivered to the 82576. A sideband WAKE# mechanism is recommended to support wake-enabled logic on mobile platforms during the L2 state. D3hot maps to L1 to support clock removal on mobile platforms.

5.1.2 PCIe Link Power States

Configuring an 82576 into D-states automatically causes the PCIe links to transition to the appropriate L-states.

- L2/L3 Ready: This link state prepares the PCIe link for the removal of power and clock. The 82576 is in the D3hot state and is preparing to enter D3cold. The power-saving opportunities for this state include, but are not limited to, clock gating of all PCIe architecture logic, shutdown of the PLL, and shutdown of all transceiver circuitry.
- L2: This link state is intended to comprehend D3cold with auxiliary power support. Note that sideband WAKE# signaling is recommended to cause wake-capable devices to exit this state. The power-saving opportunities for this state include, but are not limited to, shutdown of all transceiver circuitry except detection circuitry to support exit, clock gating of all PCIe logic, and shutdown of the PLL as well as appropriate platform voltage and clock generators.

- L3 (link off): Power and clock are removed in this link state, and there is no auxiliary power available. To bring the 82576 and its link back up, the platform must go through a boot sequence where power, clock, and reset are reapplied appropriately.
- L2/L3 ready and L2 are the link states with the lowest power but longest exit latency whereas L0 is the link state with lower power but short exit latency.

5.2 82576 Power States

The 82576 supports D0 and D3 power states defined in the PCI Power Management and PCIe specifications. D0 is divided into two sub-states: D0u (D0 un-initialized), and D0a (D0 active). In addition, the 82576 supports a Dr state that is entered when PE_RST_N is asserted (including the D3cold state).

Figure 5-1 shows the power states and transitions between them.

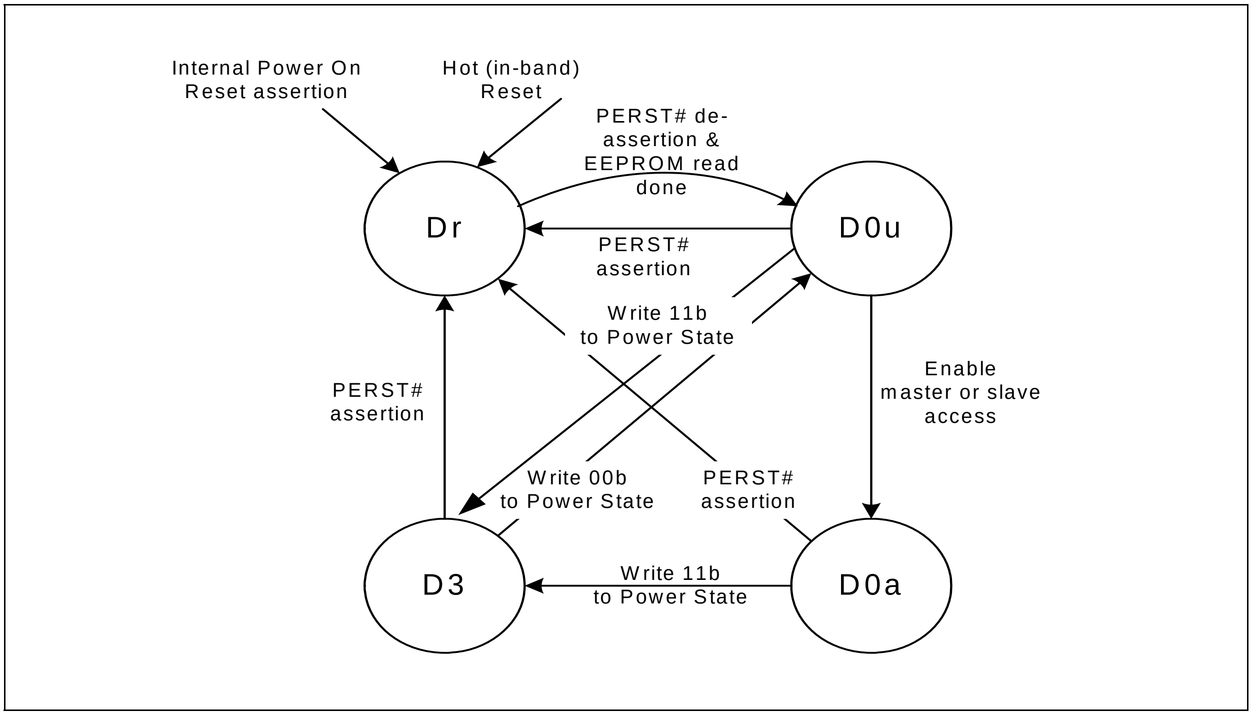


Figure 5-1. Power Management State Diagram

5.2.1 D0uninitialized State

The D0u state is a low-power state used after main PCIe reset (*PE_RST_N*) is de-asserted following power-up (cold or warm), on hot reset (in-band reset through PCIe physical layer message), or on D3 exit.

When entering D0u, the 82576 disables wake ups and asserts a reset to the PHY while the EEPROM is being read. If the *APM Mode* bit in the EEPROM's Initialization Control Word 2 is set, then APM wake up is enabled.

5.2.1.1 Entry into D0u state

D0u is reached from either the Dr state (on de-assertion of *PE_RST_N*) or the D3hot state (by configuration software writing a value of 00b to the *Power State* field of the PCI PM registers).

De-asserting *PE_RST_N* means that the entire state of the 82576 is cleared, other than sticky bits. State is loaded from the EEPROM, followed by establishment of the PCIe link. Once this is done, configuration software can access the 82576.

On a transition from D3 to D0u state, the 82576 PCI configuration space is not reset. However, the 82576 requires that software perform a full re-initialization of the function including its PCI configuration space.

5.2.2 D0active State

Once memory space is enabled, the 82576 enters an active state. It can transmit and receive packets if properly configured by the software device driver. The PHY is enabled or re-enabled by the software device driver to operate/auto-negotiate to full line speed/power if 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 (WUC) register or activate other wake-up filters by writing to the Wake Up Filter Control (WUFC) register.

5.2.2.1 Entry to D0a state

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 also enabled.

5.2.3 D3 State (PCI-PM D3hot)

The 82576 transitions to D3 when the system writes a 11b to the Power State field of the Power Management Control/Status Register (PMCSR). Any wake-up filter settings that were enabled before entering this reset state are maintained. Upon completion of or during the transition to D3 state, the 82576 clears the Memory Access Enable and I/O Access Enable bits of the PCI Command register, which disables memory access decode. In D3, the 82576 only responds to PCI configuration accesses and does not generate master cycles.

Configuration and message requests are the only TLPs accepted by a function in the D3hot state. All other received requests must be handled as unsupported requests, and all received completions can optionally be handled as unexpected completions. If an error caused by a received TLP (such as an unsupported request) is detected while in D3hot, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. See section 5.3.1.4.1 in The PCIe Base Specification

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 a 00b to the *Power State* field of the Power Management Control/Status Register (PMCSR). Transition to Dr state is through *PE_RST_N* assertion.

5.2.3.1 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 82576; it masks all interrupts and does not write to the Transmit Descriptor Tail (TDT) register or to the Receive Descriptor Tail (RDT) register and operates the master disable algorithm as defined in Section 5.2.3.2.

If wake up capability is needed, the software device driver should set up the appropriate wake up registers and then the system writes a 1b to the *PME_En* bit of the Power Management Control / Status Register (PMCSR) or to the Auxiliary (AUX) *Power PM Enable* bit of the PCIe Device Control register prior to the transition to D3.

As a response to being programmed into D3 state, the 82576 transitions its PCIe link into the L1 link state. As part of the transition into L1 state, the 82576 suspends scheduling of new TLPs and waits for the completion of all previous TLPs it has sent. The 82576 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 are kept in the 82576 (and discarded later on D3 exit). Any transmit packets that have not be sent can still be transmitted (assuming the Ethernet link is up).

In order to reduce power consumption, if the link is still needed for manageability or wake-up functionality, the PHY auto-negotiates to a lower link speed on D3 entry (see Section 3.5.7.6.4).

5.2.3.2 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 that time on, the 82576 must not issue master accesses for this function. Due to the full-duplex nature of PCIe, and the pipelined design in the 82576, it might happen that multiple requests from several functions are pending when the master disable request arrives. The protocol described in this section insures 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 82576 function and its software device driver:

- *GIO Master Disable* bit in the Device Control (CTRL) register - When the *GIO Master Disable* bit is set, the 82576 blocks new master requests by this function. The 82576 then proceeds to issue any pending requests by this function. This bit is cleared on master reset (Internal_Power_On_Reset to software reset) to enable master accesses.
- *GIO Master Enable Status* bits in the Device Status register - Cleared by the 82576 when the *GIO Master Disable* bit is set and no master requests are pending by the relevant function. Set otherwise. Indicates that no master requests are issued by this function as long as the *GIO Master Disable* bit is set. The following activities must end before the 82576 clears the *GIO Master Enable Status* bit:
  - — Master requests by the transmit and receive engines
  - — All pending completions to the 82576 are received.

Note: The software device driver sets the *GIO Master Disable* bit when notified of a pending master disable (or D3 entry). The 82576 then blocks new requests and proceeds to issue any pending requests by this function. The software device driver then polls the *GIO Master Enable Status* bit. Once the bit is cleared, it is guaranteed that no requests are pending from this function. The software device driver might time out if the *GIO Master Enable Status* bit is not cleared within a given time.

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.

5.2.4 Dr State (D3cold)

Transition to Dr state is initiated on several occasions:

- On system power up - Dr state begins with the assertion of the internal power detection circuit (*PE_RST_N*) and ends with de-assertion of *PE_RST_N*.
- On transition from a D0a state - During operation the system might assert *PE_RST_N* at any time. In an ACPI system, a system transition to the G2/S5 state causes a transition from D0a to Dr state.
- On transition from a D3 state - The system transitions the 82576 into the Dr state by asserting PCIe *PE_RST_N*.

Any wake-up filter settings that were enabled before entering this reset state are maintained.

The system might maintain *PE_RST_N* asserted for an arbitrary time. The de-assertion (rising edge) of *PE_RST_N* causes a transition to D0u state.

While in Dr state, the 82576 might enter one of several modes with different levels of functionality and power consumption. The lower-power modes are achieved when the 82576 is not required to maintain any functionality (see Section 5.2.4.1).

Note that if the 82576 is configured to provide a 50 MHz NC-SI clock (via the *NC-SI Output Clock* EEPROM bit), then the NC-SI clock must be provided in Dr state as well.

5.2.4.1 Dr Disable Mode

The 82576 enters a Dr disable mode on transition to D3cold state when it does not need to maintain any functionality. The conditions to enter either state are:

- The 82576 (all PCI functions) is in Dr state
- APM WOL is inactive for both LAN functions
- Pass-through manageability is disabled
- ACPI PME is disabled for all PCI functions
- The 82576 *Power Down En* EEPROM bit is set (default hardware value is disabled).
- The PHY *Power Down Enable* EEPROM bit is set (word 0xF, bit 6).

Entering Dr disable mode is usually done by asserting PCIe *PE_RST_N*. It might also be possible to enter Dr disable mode by reading the EEPROM while already in Dr state. The usage model for this later case is on system power up, assuming that manageability and wake up are not required. Once the 82576 enters Dr state on power-up, the EEPROM is read. If the EEPROM contents determine that the conditions to enter Dr disable mode are met, the 82576 then enters this mode (assuming that PCIe *PE_RST_N* is still asserted).

Note: Exiting Dr disable mode can be done by de-asserting PCIe *PE_RST_N*.

5.2.4.2 Entry to Dr State

Dr entry on platform power-up begins with the assertion of the internal power detection circuit (*PE_RST_N*). The EEPROM is read and determines 82576 configuration. If the *APM Enable* bit in the EEPROM's Initialization Control Word 3 is set, then APM wake up is enabled. PHY and MAC states are determined by the state of manageability and APM wake. To reduce power consumption, if manageability or APM wake is enabled, the PHY auto-negotiates to a lower link speed on Dr entry (see Section 3.5.7.6.4). The PCIe link is not enabled in Dr state following system power up (since *PE_RST_N* is asserted).

Entering Dr state from D0a state can be done by asserting *PE_RST_N*. An ACPI transition to the G2/S5 state is reflected in an 82576 transition from D0a to Dr state. The transition can be orderly (such as, user selected the shut down option), in which case the software device driver might have a chance to intervene. Or, it might be an emergency transition (such as power button override), in which case, the software device driver is not notified.

To reduce power consumption, if any of manageability, APM wake or PCI-PM PME[1] is enabled, the PHY auto-negotiates to a lower link speed on D0a to Dr transition (see Section 3.5.7.6.4).

Transition from D3(hot) state to Dr state is done by asserting PE_RST_N. Prior to that, the system initiates a transition of the PCIe link from L1 state to either the L2 or L3 state (assuming all functions were already in D3 state). The link enters L2 state if PCI-PM PME is enabled.

5.2.4.3 Auxiliary Power Usage

If DisableD3Cold=0b, the 82576 uses the AUX_PWR indication that auxiliary power is available to it, and therefore advertises D3cold wake up support. The amount of power required for the function (including the entire NIC) is advertised in the Power Management Data register, which is loaded from the EEPROM.

If D3cold is supported, the *PME_En* and *PME_Status* bits of the Power Management Control/Status Register (PMCSR), as well as their shadow bits in the Wake Up Control (WUC) register are reset only by the power up reset (detection of power rising).

The only effect of setting *AUX_PWR* to 1b is advertising D3cold wake-up support and changing the reset function of *PME_En* and *PME_Status*. Note that *AUX_PWR* is a strapping option in the 82576.

5.2.5 Link Disconnect

In any of D0u, D0a, D3, or Dr power states, the 82576 enters a link-disconnect state if it detects a link-disconnect condition on the Ethernet link. Note that the link-disconnect state is invisible to software (other than the *Link Energy Detect* bit state). In particular, while in D0 state, software might be able to access any of the 82576 registers as in a link-connect state.

5.2.6 Device Power-Down State

The 82576 enters a global power-down state if all of the following conditions are met:

- The 82576 *Power Down Enable* EEPROM bit (word 0x1E bit 15) was set (default hardware value is disabled).
- The 82576 is in Dr state.
- The link connections of both ports (PHY or SerDes) are in power down mode.

The 82576 also enters a power-down state when the DEV_OFF_N pin is active and the relevant EEPROM bits were configured as previously described (see Section 4.4 for more details on DEV_OFF_N functionality).

5.3 Power Limits by Certain Form Factors

The 82576 exceeds the allocated auxiliary power in some configurations (such as both ports running at 1000 Mb/s speed). The 82576 must therefore be configured to meet requirements. To do so, the 82576 implements three EEPROM bits to disable operation in certain cases:

[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."

1. The *Disable_1000* PHY register bit disables 1000 Mb/s operation under all conditions.
2. The Disable *1000 in non-D0a* PHY CSR bit disables 1000 Mb/s operation in non-D0a states[1]. If *Disable 1000 in non-D0a* is set, and the 82576 is at 1000 Mb/s speed on entry to a non-D0a state, then the 82576 removes advertisement for 1000 Mb/s and auto-negotiates.

Note that the 82576 restarts link auto-negotiation each time it transitions from a state where 1000 Mb/s or 100 Mb/s speed is enabled to a state where 1000 Mb/s or 100 Mb/s speed is disabled, or vice versa. For example, if *Disable 1000 in non-D0a* is set but *Disable_1000* is cleared, the 82576 restarts link auto-negotiation on transition from D0 state to D3 or Dr states.

5.4 Interconnects Power Management

This section describes the power reduction techniques employed by the 82576 main interconnects.

5.4.1 PCIe* Link Power Management

The PCIe link state follows the power management state of the 82576. Since the 82576 incorporates multiple PCI functions, its power management state is defined as the power management state of the most awake function (see Figure 5-2):

- If any function is in D0 state (either D0a or D0u), the PCIe link assumes the 82576 is in D0 state. Else,
- If the functions are in D3 state, the PCIe link assumes the 82576 is in D3 state. Else,
- The 82576 is in Dr state (PE_RST_N is asserted to all functions).

The 82576 supports all PCIe power management link states:

- L0 state is used in D0u and D0a states.
- The L0s state is used in D0a and D0u states each time link conditions apply.
- The L1 state is also used in D0a and D0u states when idle conditions apply for a longer period of time. The L1 state is also used in the D3 state.
- The L2 state is used in the Dr state following a transition from a D3 state if *PCI-PM PME* is enabled.
- The L3 state is used in the Dr state following power up, on transition from D0a, and if *PME* is not enabled in other Dr transitions.

The 82576's support for active state link power management is reported via the PCIe Active State Link PM Support register and is loaded from the EEPROM.

[1] The restriction is defined for all non-D0a states to have compatible behavior with previous products.

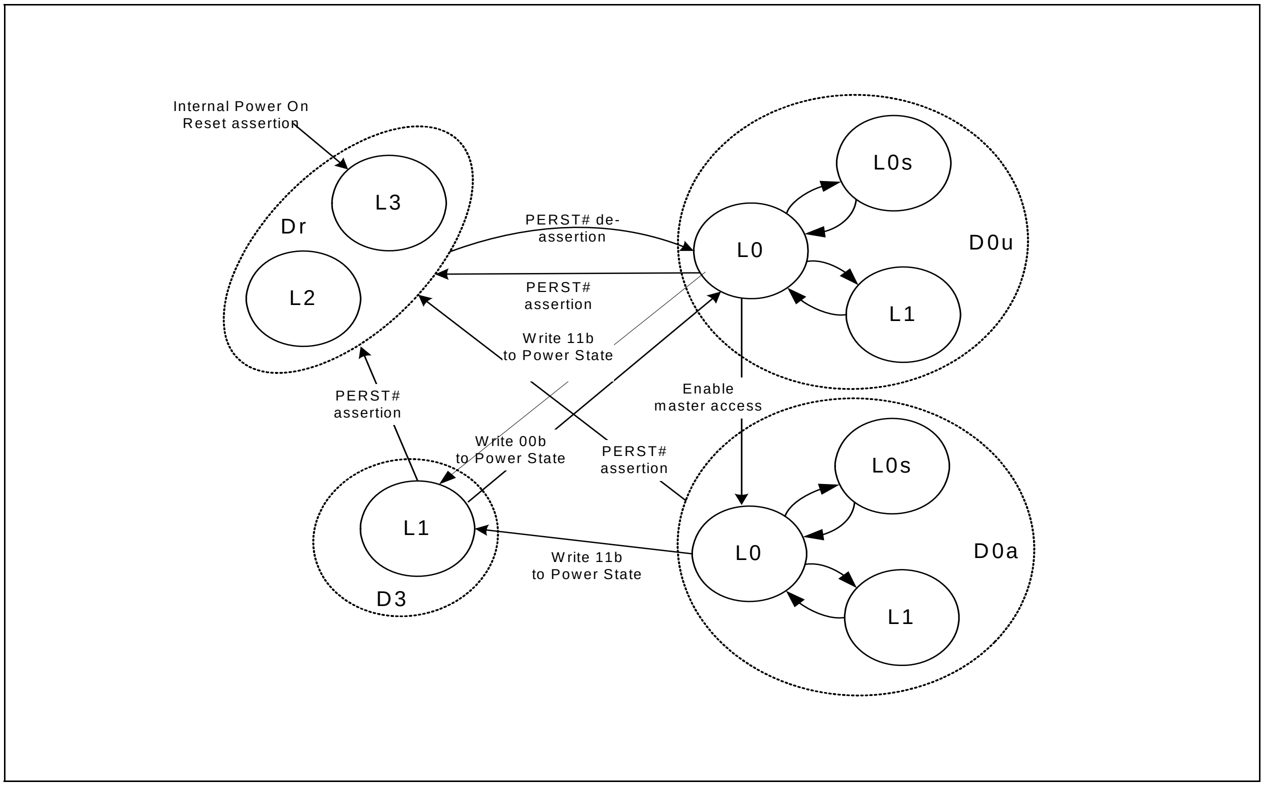


Figure 5-2. Link Power Management State Diagram

While in L0 state, the 82576 transitions the transmit lane(s) into L0s state once the idle conditions are met for a period of time as follows:

L0s configuration fields are:

- L0s enable - The default value of the *Active State Link PM Control* field in the PCIe Link Control register is set to 00b (both L0s and L1 disabled). System software might later write a different value into the PCIe Link Control register. The default value is loaded on any reset of the PCI configuration registers.
- The *L0S_ENTRY_LAT* bit in the PCIe Control Register (GCR), determines L0s entry latency. When set to 0b, L0s entry latency is the same as L0s exit latency of the 82576 at the other end of the link. When set to 1b, L0s entry latency is 1/4 of the L0s exit Latency of the 82576 at the other end of the link. The default value is 0b (entry latency is the same as L0s exit latency of the 82576 at the other end of the link).
- L0s exit latency (as published in the *L0s Exit Latency* field of the Link Capabilities register) is loaded from EEPROM. Separate values are loaded when the 82576 shares the same reference PCIe clock with its partner across the link, and when the 82576 uses a different reference clock than its partner across the link. The 82576 reports whether it uses the slot clock configuration, through the PCIe *Slot Clock Configuration* bit loaded from the *Slot_Clock_Cfg* EEPROM bit.
- L0s acceptable latency (as published in the *Endpoint L0s Acceptable Latency* field of the Device Capabilities register) is loaded from EEPROM.

L1 configuration fields are:

- L1 entry latency — The 82576 enters the L1 state after it has been in the L0s state (in both directions) for a period of time determined by the Latency_To_Enter_L1 CSR register. The initial value is loaded from the *Latency_To_Enter_L1* EEPROM field.
- L1 exit latency (as published in the *L1 Exit Latency* field of the Link Capabilities register) is loaded from the *L1_Act_Ext_Latency Latency_To_Enter_L1* field in the EEPROM.
- L1 acceptable latency (as published in the *Endpoint L1 Acceptable Latency* field of the Device Capabilities register) is loaded from EEPROM.

5.4.2 NC-SI Clock Control

The 82576 can be configured to provide a 50 MHz output clock to its NC-SI interface and other platform devices. When enabled(through the NC-SI *Output Clock* EEPROM bit), the NC-SI clock is provided in all power states without exception.

5.4.3 PHY Power-Management

THe PHY power management features are described in Section 3.5.7.6.

5.4.4 SerDes/SGMII Power Management

Each 82576 SerDes enters a power-down state when none of its clients is enabled and therefore has no need to maintain a link. This can happen in one of the following cases. Note that SerDes power-down must be enabled through the EEPROM SerDes *Low Power Enable* bit.

1. D3/Dr state: Each SerDes enters a low-power state if the following conditions are met:
   a. The LAN function associated with this SerDes is in a non-D0 state
   b. APM WOL is inactive
   c. Pass-through manageability is disabled
   d. ACPI PME is disabled
2. PHY mode: Each SerDes is disabled when its LAN function is configured to PHY mode.
3. LAN disable: Each SerDes can be disabled if its LAN function's LAN Disable input indicates that the relevant function should be disabled. Since the SerDes is shared between the LAN function and manageability, it might not be desired to power down the SerDes in LAN Disable. The *PHY_in_LAN_Disable* EEPROM bit determines whether the SerDes is powered down when the LAN Disable pin is asserted. The default is not to power down.

5.5 Timing of Power-State Transitions

The following sections give detailed timing for the state transitions. In the diagrams the dotted connecting lines represent the 82576 requirements, while the solid connecting lines represent the 82576 guarantees.

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

5.5.1 Power Up (Off to Dup to D0u to D0a

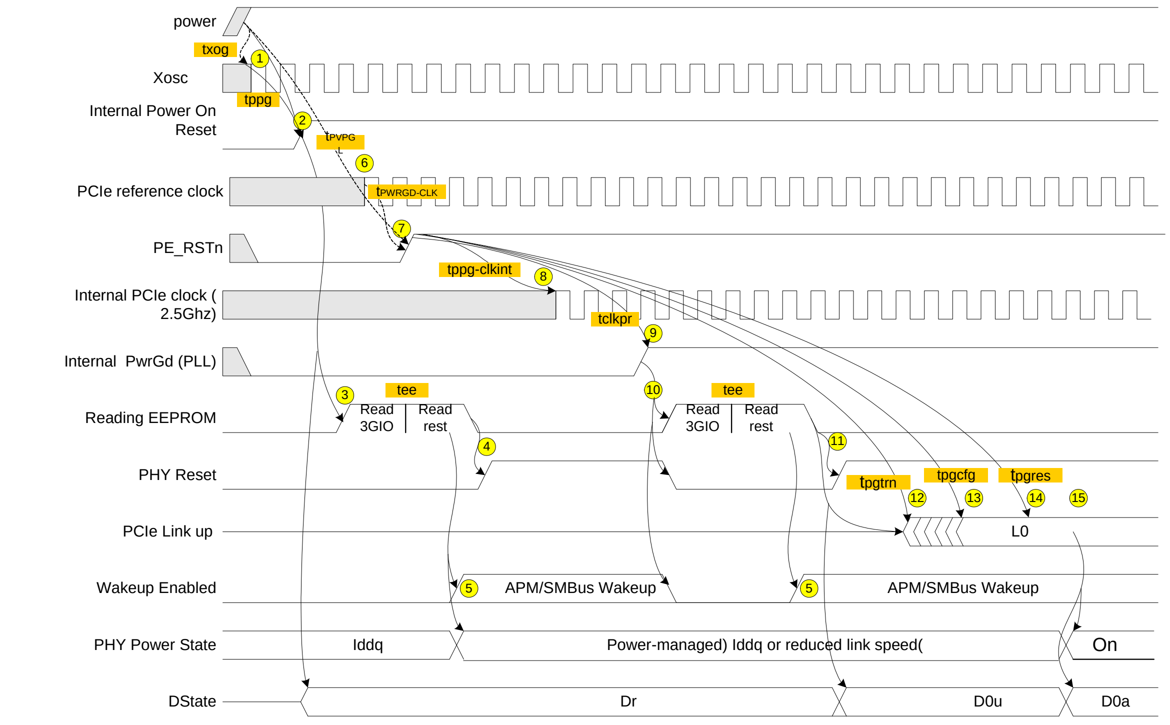


Figure 5-3. Power Up (Off to Dup to D0u to D0a)

Table 5-1. Power Up (Off to Dup to D0u to D0a)

Note	Description
1	Xosc is stable t_{xog} after power is stable.
2	Internal_Power_On_Reset 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 Internal_Power_On_Reset.
4	After reading the EEPROM, PHY reset is de-asserted.
5	APM wake-up mode can be enabled based on what is read from the EEPROM.
6	The PCIe reference clock is valid $t_{PE\_RST-CLK}$ before de-asserting PE_RST_N (according to PCIe specification).
7	PE_RST_N is de-asserted t_{PVPGL} after power is stable (according to PCIe specification).
8	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
9	The PCIe internal PWRGD signal is asserted t_{clkpr} after the external PE_RST_N signal.
10	Asserting internal PCIe PWRGD causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
11	After reading the EEPROM, PHY reset is de-asserted.
12	Link training starts after t_{pgtrn} from PE_RST_N de-assertion.

Table 5-1. Power Up (Off to Dup to D0u to D0a) (Continued)

13	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST_N de-assertion.
14	A first PCI configuration response can be sent after t_{pgres} from PE_RST_N de-assertion.
15	Writing a 1b to the *Memory Access Enable* bit in the PCI Command Register transitions the 82576 from D0u to D0. state.

5.5.2 Transition from D0a to D3 and Back Without PE_RST_N

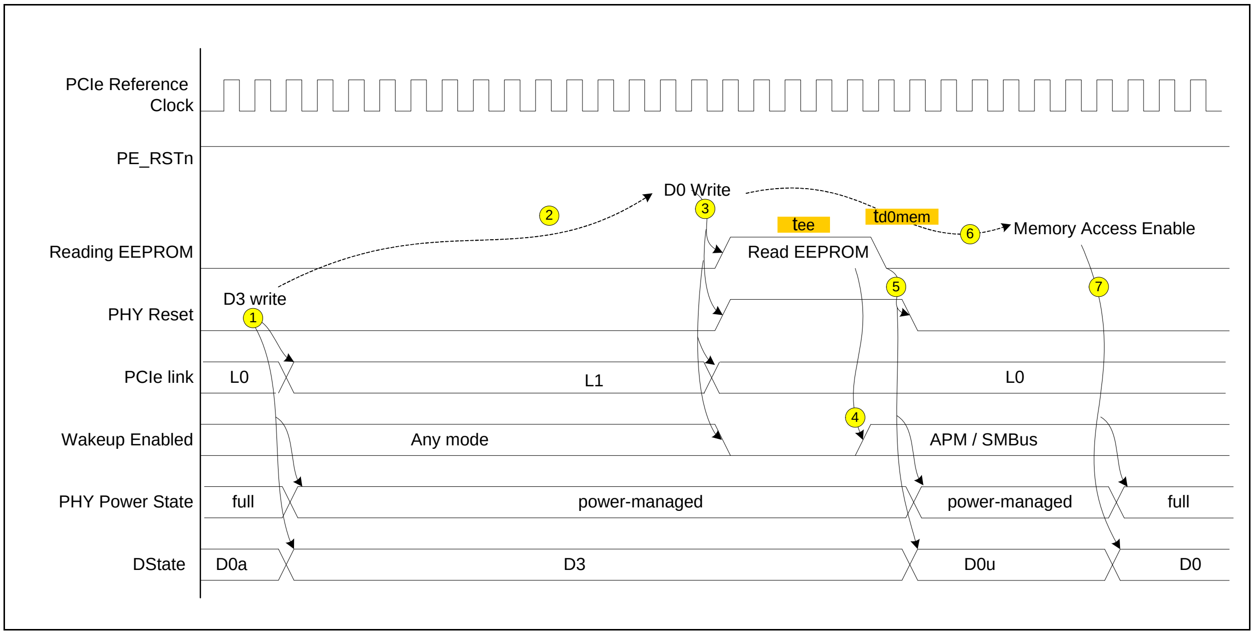


Figure 5-4. Transition from D0a to D3 and Back Without PE_RST_N

Table 5-2. Transition from D0a to D3 and Back Without PE_RST_N

Note	Description
1	Writing 11b to the *Power State* field of the Power Management Control/Status Register (PMCSR) transitions the 82576 to D3.
2	The system can keep the 82576 in D3 state for an arbitrary amount of time.
3	To exit D3 state, the system writes 00b to the *Power State* field of the PMCSR.
4	APM wake-up or SMBus mode might be enabled based on what is read in the EEPROM.
5	After reading the EEPROM, reset to the PHY is de-asserted. The PHY operates at reduced-speed if APM wake up or SMBus is enabled, else 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 82576 from D0u to D0 state and returns the PHY to full-power/speed operation.

5.5.3 Transition From D0a to D3 and Back With PE_RST_N

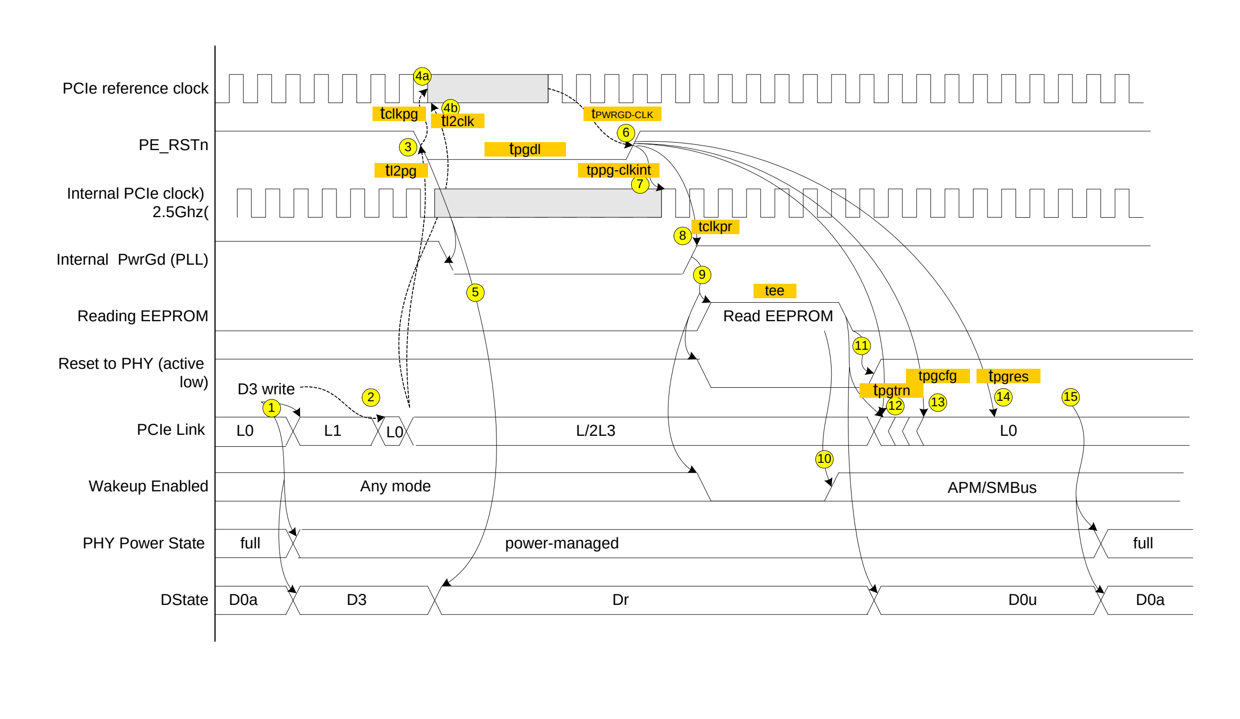


Figure 5-5. Transition From D0a to D3 and Back With PE_RST_N

Table 5-3. Transition From D0a to D3 and Back With PE_RST_N

Note	Description
1	Writing 11b to the *Power State* field of the PMCSR transitions the 82576 to D3. PCIe link transitions to L1 state.
2	The system can delay an arbitrary amount of time between setting D3 mode and transitioning the link to an L2 or L3 state.
3	Following link transition, PE_RST_N is asserted.
4	The system must assert PE_RST_N before stopping the PCIe reference clock. It must also wait t_{l2clk} after link transition to L2/L3 before stopping the reference clock.
5	On assertion of PE_RST_N, the 82576 transitions to Dr state.
6	The system starts the PCIe reference clock $t_{PE\_RST-CLK}$ before de-assertion PE_RST_N.
7	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
8	The PCIe internal PWRGD signal is asserted t_{clkpr} after the external PE_RST_N signal.
9	Asserting internal PCIe PWRGD causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
10	APM wake-up mode might be enabled based on what is read from the EEPROM.
11	After reading the EEPROM, PHY reset is de-asserted.
12	Link training starts after t_{pgtrn} from PE_RST_N de-assertion.
13	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST_N de-assertion.
14	A first PCI configuration response can be sent after t_{pgres} from PE_RST_N de-assertion.
15	Writing a 1b to the *Memory Access Enable* bit in the PCI Command Register transitions the 82576 from D0u to D0 state.

5.5.4 Transition From D0a to Dr and Back Without Transition to D3

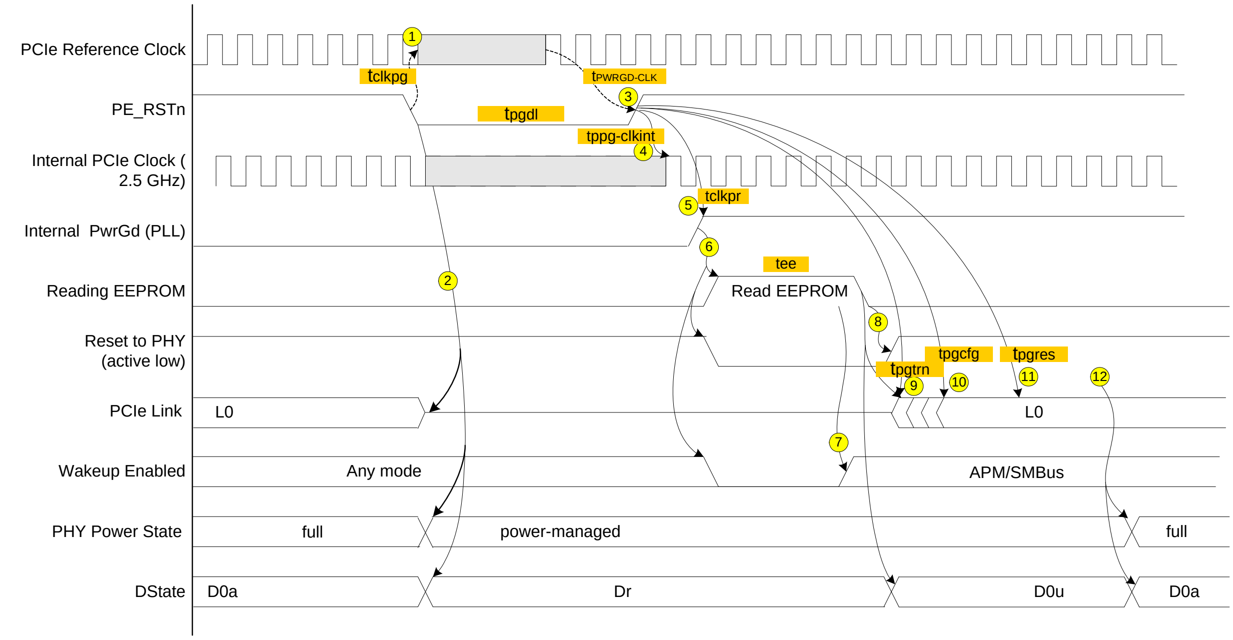


Figure 5-6. Transition From D0a to Dr and Back Without Transition to D3

Table 5-4. Transition From D0a to Dr and Back Without Transition to D3

Note	Description
1	The system must assert PE_RST_N before stopping the PCIe reference clock. It must also wait t_{l2clk} after link transition to L2/L3 before stopping the reference clock.
2	On assertion of PE_RST_N, the 82576 transitions to Dr state and the PCIe link transition to electrical idle.
3	The system starts the PCIe reference clock $t_{PE\_RST-CLK}$ before de-assertion PE_RST_N.
4	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
5	The PCIe internal PWRGD signal is asserted t_{clkpr} after the external PE_RST_N signal.
6	Asserting internal PCIe PWRGD causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
7	APM wake-up mode might be enabled based on what is read from the EEPROM.
8	After reading the EEPROM, PHY reset is de-asserted.
9	Link training starts after t_{pgtrn} from PE_RST_N de-assertion.
10	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST_N de-assertion.
11	A first PCI configuration response can be sent after t_{pgres} from PE_RST_N de-assertion.
12	Writing a 1b to the *Memory Access Enable* bit in the PCI Command Register transitions the 82576 from D0u to D0 state.

5.6 Wake Up

The 82576 supports two modes of wake-up management:

1. Advanced Power Management (APM) wake up
2. ACPI/PCIe defined wake up

The usual model is to activate one mode at a time but not both modes together. If both modes are activated, the 82576 might wake up the system in unexpected events. For example, if APM is enabled together with PCIe PME, a magic packet might wake up the system even if APMPME is disabled. Alternatively, if APM is enabled together with some PCIe filters, packets matching these filters might wake up the system even if PCIe PME is disabled.

5.6.1 Advanced Power Management Wake Up

Advanced Power Management Wake Up or APM Wakeup (also known as Wake on LAN) is a feature that existed in earlier 10/100 Mb/s NICs. This functionality was designed to receive a broadcast or unicast packet with an explicit data pattern, and then assert a subsequent signal to wake up the system. This was accomplished by using a special signal that ran across a cable to a defined connector on the motherboard. The NIC would assert the signal for approximately 50 ms to signal a wake up. The 82576 now uses (if configured) an in-band PM_PME message for this functionality.

On power up, the 82576 reads the *APM Enable* bits from the EEPROM Initialization Control Word 3 into the *APM Enable* (APME) bits of the Wakeup Control (WUC) register. These bits control enabling of APM wake up.

When APM wake up is enabled, the 82576 checks all incoming packets for Magic Packets. See Section 5.6.3.1.4 for a definition of Magic Packets.

Once the 82576 receives a matching magic packet, and if the *Assert PME On APM Wakeup* (APMPME) bit is set in the Wake Up Control (WUC) register, it:

- Sets the *PME_Status* bit in the PMCSR and issues a PM_PME message (in some cases, this might require asserting the WAKE# signal first to resume power and clock to the PCIe interface).
- Stores the first 128 bytes of the packet in the Wake Up Packet Memory (WUPM) register.
- Sets the *Magic Packet Received* bit in the Wake Up Status (WUS) register.
- Sets the packet length in the Wake Up Packet Length (WUPL) register.

The 82576 maintains the first Magic Packet received in the Wake Up Packet Memory (WUPM) register until the software device driver writes a 1b to the *Magic Packet Received MAG* bit in the Wake Up Status (WUS) register.

APM wake up is supported in all power states and only disabled if a subsequent EEPROM read results in the *APM Wake Up* bit being cleared or software explicitly writes a 0b to the *APM Wake Up* (APM) bit of the WUC register.

5.6.2 PCIe Power Management Wake Up

The 82576 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 wakeup packet.
- Detection of a link change of state.

Activating PCIe power management wake up requires the following:

- The software device driver programs the Wake Up Filter Control (WUFC) register to indicate the packets it needs to wake up and supplies the necessary data to the IPv4/v6 Address Table (IP4AT, IP6AT) and the Flexible Host Filter Table (FHFT). It can also set the Link Status Change Wake Up

Enable (LNKC) bit in the Wake Up Filter Control (WUFC) register to cause wake up when the link changes state.

- The operating system (at configuration time) writes a 1b to the PME_En bit of the Power Management Control/Status (PMCSR.8) register.

Normally, after enabling wake up, the operating system write a 11b to the lower two bits of the PMCSR to put the 82576 into low-power mode.

Once wake up is enabled, the 82576 monitors incoming packets, first filtering them according to its standard address filtering method, then filtering them with all of the enabled wakeup filters. If a packet passes both the standard address filtering and at least one of the enabled wakeup filters, the 82576:

- Sets the *PME_Status* bit in the PMCSR.
- Asserts PE_WAKE_N (if the *PME_En* bit in the PMCSR is set).
- Stores the first 128 bytes of the packet in the Wakeup Packet Memory (WPM) register.
- Sets one or more of the received bits in the Wake Up Status (WUS) register. Note that the 82576 sets more than one bit if a packet matches more than one filter.
- Sets the packet length in the Wake Up Packet Length (WUPL) register.

If enabled, a link state change wake up causes similar results, setting *PME_Status*, asserting *PE_WAKE_N* and setting the *Link Status Changed* (LNKC) bit in the Wake Up Status (WUS) register when the link goes up or down.

The 82576 supports the following change described in the PCIe Base Specification, Rev. 1.1RD (section 5.3.3.4) - On receiving a PME_Turn_Off message, the 82576 must block the transmission of PM_PME messages and transmit a PME_TO_Ack message upstream. The 82576 is permitted to send a PM_PME message after the Link is returned to an L0 state through LDn.

PE_WAKE_N remains asserted until the operating system either writes a 1b to the PME_Status bit of the PMCSR register or writes a 0b to the PME_En bit.

After receiving a wake-up packet, the 82576 ignores any subsequent wake-up packets until the software device driver clears all of the received bits in the Wake Up Status (WUS) register. It also ignores link change events until the software device driver clears the Link Status Changed (LNKC) bit in the Wake Up Status (WUS) register.

Note: A wake on link change is not supported when configured to SerDes mode.

5.6.3 Wake-Up Packets

The 82576 supports various wake-up 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 (WUFC) register is set to 1b.

5.6.3.1 Pre-Defined Filters

The following packets are supported by the 82576's pre-defined filters:

- Directed packet (including exact, multicast indexed, 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 Wakeup Filter Control (WUFC) register is set to 1b.

The explanation of each filter includes a table showing which bytes at which offsets are compared to determine if the packet passes the filter.

Note: Both VLAN frames and LLC/SNAP can increase the given offsets if they are present.

5.6.3.1.1 Directed Exact Packet

The 82576 generates a wake-up event after receiving any packet whose destination address matches one of the 24 valid programmed receive addresses, if the *Directed Exact Wake Up Enable* bit is set in the Wake Up Filter Control (WUFC.EX) register.

5.6.3.1.2 Directed Multicast Packet

For multicast packets, the upper bits of the incoming packet's destination address index a bit vector, the Multicast Table Array (MTA) that indicates whether to accept the packet. If the *Directed Multicast Wake Up Enable* bit set in the Wake Up Filter Control (WUFC.MC) register and the indexed bit in the vector is one, then the 82576 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 (RCTL.MO) register.

5.6.3.1.3 Broadcast

5.6.3.1.4 If the *Broadcast Wake Up Enable* bit in the Wake Up Filter Control (WUFC.BC) register is set, **the 82576** generates a wake-up event when it receives a broadcast packet.**Magic Packet**

Magic packets are defined in:

http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/20213.pdf as:

“Once the LAN controller has been put into the Magic Packet mode, it scans all incoming frames addressed to the node for a specific data sequence.This sequence indicates to the controller that this is a Magic Packet frame. A Magic Packet frame must also meet the basic requirements for the LAN technology chosen, such as SOURCE ADDRESS, DESTINATION ADDRESS (which may be the receiving station's IEEE address or a MULTICAST address which includes the BROADCAST address), and CRC. The specific data sequence consists of 16 repetitions of the IEEE address of this node, with no breaks or interruptions. This sequence can be located anywhere within the packet, but must be preceded by a synchronization stream. The synchronization stream allows the scanning state machine to be much simpler. The synchronization stream is defined as 6 bytes of 0xFF. The device will also accept a BROADCAST frame, as long as the 16 repetitions of the IEEE address match the address of the machine to be awakened.”

The 82576 expects the destination address to either:

- Be the broadcast address (FF.FF.FF.FF.FF.FF)
- Match the value in Receive Address 0 (RAH0, RAL0) register. This is initially loaded from the EEPROM but can be changed by the software device driver.
- Match any other address filtering enabled by the software device driver.

The 82576 searches for the contents of Receive Address 0 (RAH0, RAL0) register as the embedded IEEE address. It considers any non-0xFF byte after a series of at least 6 0xFFs to be the start of the IEEE address for comparison purposes. For example, it catches the case of 7 0xFFs followed by the IEEE address). As soon as one of the first 96 bytes after a string of 0xFFs don't match, it continues to search for anther set of at least 6 0xFFs followed by the 16 copies of the IEEE address later in the packet. Note that this definition precludes the first byte of the destination address from being FF.

A Magic Packet's destination address must match the address filtering enabled in the configuration registers with the exception that broadcast packets are considered to match even if the *Broadcast Accept* bit of the Receive Control (RCTL.BAM) register is 0b. If APM wake up (wake up by a Magic Packet) is enabled in the EEPROM, the 82576 starts up with the Receive Address 0 (RAH0, RAL0) register loaded from the EEPROM. This enables the 82576 to accept packets with the matching IEEE address before the software device driver loads.

Table 5-5. Magic Packet Structure

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	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 0 (RAH0, RAL0) register.

Note: Accepting broadcast Magic Packets for wake up purposes when the *Broadcast Accept* bit of the Receive Control (RCTL.BAM) register is 0b is a change from a previous device, which initialized RCTL.BAM to 1 if APM was enabled in the EEPROM, but then required that bit to be 1b to accept broadcast Magic Packets, unless broadcast packets passed another perfect or multicast filter.

5.6.3.1.5 ARP / IPv4 Request Packet

The 82576 supports receiving ARP request packets for wake up if the *ARP* bit is set in the Wake Up Filter Control (WUFC) register. Four IPv4 addresses are supported, which are programmed in the IPv4 Address Table (IP4AT). A successfully matched packet must contain a broadcast MAC address, a protocol type of 0x0806, an ARP op-code of 0x01, and one of the four programmed IPv4 addresses. The 82576 also handles ARP request packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-6. ARP Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0x0806	Compare	ARP
14	2	HW Type	0x0001	Compare	
16	2	Protocol Type	0x0800	Compare	

Table 5-6. ARP Packet Structure and Processing

18	1	Hardware Size	0x06	Compare	
19	1	Protocol Address Length	0x04	Compare	
20	2	Operation	0x0001	Compare	
22	6	Sender HW Address	-	Ignore	
28	4	Sender IP Address	-	Ignore	
32	6	Target HW Address	-	Ignore	
38	4	Target IP Address	IP4AT	Compare	May match any of four values in IP4AT.

5.6.3.1.6 Directed Ipv4 Packet

The 82576 supports receiving directed IPv4 packets for wake up if the *IPV4* bit is set in the Wake Up Filter Control (WUFC) register. Four IPv4 addresses are supported, which are programmed in the IPv4 Address Table (IP4AT). A successfully matched packet must contain the station's MAC address, a protocol type of 0x0800, and one of the four programmed IPv4 addresses. The 82576 also handles directed IPv4 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-7. IPv4 Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0x0800	Compare	IP
14	1	Version/ HDR length	0x4X	Compare	Check IPv4
15	1	Type of Service	-	Ignore	
16	2	Packet Length	-	Ignore	
18	2	Identification	-	Ignore	
20	2	Fragment Info	-	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 four values in IP4AT.

5.6.3.1.7 Directed IPv6 Packet

The 82576 supports receiving directed IPv6 packets for wake up if the *IPV6* bit is set in the Wake Up Filter Control (WUFC) register. One IPv6 address is supported and is programmed in the IPv6 Address Table (IP6AT). A successfully matched packet must contain the station's MAC address, a protocol type of 0x86DD, and the programmed IPv6 address. In addition, the *IPAV.V60* bit should be set. The 82576 also handles directed IPv6 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-8. IPv6 Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0x86DD	Compare	IP
14	1	Version/ Priority	0x6X	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.

5.6.3.2 Flexible Filters

The 82576 supports a total of six flexible filters. Each filter is can be configured to recognize any arbitrary pattern within the first 128 bytes of the packet. To configure the flexible filters, software programs the mask values (required values and the minimum packet length), into the Flexible Host Filter Table (FHFT and FHFT_EXT). These six flexible filters contain separate values for each filter. Software must also enable the filters in the Wake Up Filter Control (WUFC) register, and enable the overall wake up functionality. The overall wake up functionality must be enabled by setting PME_En in the PMCSR or the Wake Up Control (WUC) register.

Once enabled, the flexible filters scan incoming packets for a match. If the filter encounters any byte in the packet where the mask bit is one and the byte doesn't match the value programmed in the Flexible Host Filter Table (FHFT), then the filter fails that packet. If the filter reaches the required length without failing the packet, it passes the packet and generates a wake-up event. It ignores any mask bits set to one beyond the required length.

Note: The flex filters are temporarily disabled when read from or written to by the host. Any packet received during a read or write operation is dropped. Filter operation resumes once the read or write access completes.

The following packets are listed for reference purposes only. The flexible filter could be used to filter these packets.

5.6.3.2.1 IPX Diagnostic Responder Request Packet

An IPX diagnostic responder request packet must contain a valid MAC address, a protocol type of 0x8137, and an IPX diagnostic socket of 0x0456. It might include LLC/SNAP headers and VLAN tags. Since filtering this packet relies on the flexible filters, which use offsets specified by the operating system directly, the operating system must account for the extra offset LLC/SNAP headers and VLAN tags.

Table 5-9. IPX Diagnostic Responder Request Packet Structure and Processing

Offset	# 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	
12	2	Type	0x8137	Compare	IPX
14	16	Some IPX Stuff	-	Ignore	
30	2	IPX Diagnostic Socket	0x0456	Compare	

5.6.3.2.2 Directed IPX Packet

A valid directed IPX packet contains the station's MAC address, a protocol type of 0x8137, and an IPX node address that is equal to the station's MAC address. It might include LLC/SNAP headers and VLAN tags. Since filtering this packet relies on the flexible filters, which use offsets specified by the operating system directly, the operating system must account for the extra offset LLC/SNAP headers and VLAN tags.

Table 5-10. IPX Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0x8137	Compare	IPX
14	10	Some IPX Info	-	Ignore	
24	6	IPX Node Address	Receive Address 0	Compare	Must match receive address 0.

5.6.3.2.3 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.

5.6.3.3 Wake Up Packet Storage

The 82576 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) register after the system wakes up.

§ §

6. Non-Volatile Memory Map - EEPROM

6.1 EEPROM General Map

Table 6-1 lists the 82576 EEPROM map.

Table 6-1. EEPROM Map

#	Used By	High Byte	Low Byte	Which LAN
0	HW	Ethernet Address Byte 2	Ethernet Address Byte 1	Both
01	HW	Ethernet Address Byte 4	Ethernet Address Byte 3	Both
02	HW	Ethernet Address Byte 6	Ethernet Address Byte 5	Both
03	SW	Compatibility High	Compatibility Low	Both
04	SW	Compatibility High	Compatibility Low	Both
05	SW	Compatibility High	Compatibility Low	Both
06	SW	Compatibility High	Compatibility Low	Both
07	SW	Compatibility High	Compatibility Low	Both
08	SW	PBA, Byte 1	PBA, Byte 2	
09	SW	PBA, Byte 3	PBA, Byte 4	
0A	HW	Init Control 1		Both
0B	HW	Subsystem ID		Both
0C	HW	Subsystem Vendor ID		Both
0D	HW	Device ID		LAN 0
0E	HW	Reserved		
0F	HW	Init Control 2		Both
10	HW	Software Defined Pins Control		LAN 1
11	HW	Device ID		LAN 1
12	HW	EEPROM Sizing and Protected Fields		Both
13	HW	Reserved		Both
14	HW	Init Control 3		LAN 1
15	HW	PCIe Completion Timeout Configuration		Both
16	HW	MSI-X Configuration		Both
17	FW	Analog blocks pointers Address		Both
18	HW	PCIe Init Configuration 1		Both
19	HW	PCIe Init Configuration 2		Both
1A	HW	PCIe Init Configuration 3		Both
1B	HW	PCIe Control		Both
1C	HW	LEDCTL 1 3 Default		LAN 0
1D	HW	Dummy Function Device ID		Both

Table 6-1. EEPROM Map (Continued)

1E	HW	Device REV ID	Both
1F	FW	LEDCTL 0 2 Default	LAN 0
20	HW	Software Defined Pins Control	LAN 0
21	HW	Functions Control	Both
22	HW	LAN Power Consumption	Both
23	HW	Management Hardware Config Control	Both
24	HW	Init Control 3	LAN 0
25	HW	I/O Virtualization (IOV) Control	Both
26	HW	IOV Device ID	Both
27	HW	Reserved	LAN 0
28	HW	Reserved	LAN 1
29	HW	Reserved	Both
2A	HW	LEDCTL 1 - 3 Default	LAN 1
2B	HW	LEDCTL 0 - 2 Default	LAN 1
2C	HW	End of Read Only Area	Both
2D	HW	Start of Read Only Area	Both
2E	HW	Watchdog Configuration	Both
2F	OEM	VPD Pointer	
30	PXE	PXE main setup options (Primary)	
31	PXE	PXE Configuration Customization options (Primary)	
32	PXE	PXE Word (Software Use) PXE Version	
33	PXE	PXE Capabilities	
34	PXE	PXE main setup options (Secondary)	
35	PXE	PXE Configuration Customization options (Secondary)	
36	PXE	PXE Word	
37	PXE	Pointer to Alternate MAC address	
38:3C	PXE	PXE Words	
3D	PXE	iSCSI Boot Configuration Start Address	
3E	PXE	PXE Word	
3F	PXE	Software Checksum, Words 00h - 3Fh	
40	HW	NC-SI Arbitration Enable.	
40:41	HW	Reserved	
42	SW	Image Unique ID Low	
43	SW	Image Unique ID High	
44:4F	HW	Reserved	
50:53	FW	Common Firmware Pointers	MNG
54	FW	MNG Capabilities	MNG
55:5B	FW	Pass Through Pointers	MNG
5C -	FW	Firmware Data Structures	MNG

6.2 Hardware Accessed Words

This section describes the EEPROM words that are loaded by 82576 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 0x12) is valid.

6.2.1 Ethernet Address (Words 0x00-0x02)

The Ethernet Individual Address (IA) is a 6-byte field that must be unique for each NIC, and thus unique for each copy of the EEPROM image. The first three bytes are vendor specific. For example, the IA is equal to [00 AA 00] or [00 A0 C9] for Intel products. The value from this field is loaded into the Receive Address Register 0 (RAL0/RAH0).

For the purpose of this specification, the IA byte numbering convention is indicated as follows:

	IA Byte / Value					
Vendor	1	2	3	4	5	6
Intel Original	00	AA	00	variable	variable	variable
Intel New	00	A0	C9	variable	variable	variable

The Ethernet address is loaded for LAN0 and bit 41 (8th MSB) is inverted for LAN1 (bit 0 byte 6 in the EEPROM = bit 8 in EEPROM word 0x2).

6.2.2 Initialization Control Word 1 (word 0x0A)

The first word read by the 82576 contains initialization values that:

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

Bit	Name	Default in EEPROM less mode	Description
15:13	Reserved		Reserved - must be zero
12	Reserved		Reserved - must be zero
11	FRCSPD	0b	Default setting for the *Force Speed* bit in the Device Control register (CTRL[11]). See Section 8.2.1.
10	FD	0b	Default setting for duplex setting. Mapped to CTRL[0]. See Section 8.2.1.
9	Reserved	1b	Reserved - should be set to zero.
8:7	Reserved		Reserved - must be zero
6	SDP_IDDQ_ EN	0b	When set, SDP keeps their value and direction when the 82576 enters dynamic IDDQ mode. Otherwise, SDP moves to HighZ + pull-up mode in dynamic IDDQ mode. Reflected in EEDIAG (See Section 8.4.5).
5	Deadlock Timeout Enable	1b	If set, a device granted access to the EEPROM or Flash that does not toggle the interface for more than 1 second might have the grant revoked. See Section 8.2.1.
4	ILOS	0b	Default setting for the loss-of-signal polarity setting for CTRL[7]. Section 8.2.1.

3	Power Management	1b	0b = Reserved - must be one. 1b = Full support for power management (For normal operation, this bit must be set to 1b). Must be one for normal power management operation. See Section 9.5.1.
2	Reserved		Reserved - must be zero
1	Load Subsystem IDs	1b	This bit, when set to 1b, indicates that the 82576 is to load its PCIe Subsystem ID and Subsystem Vendor ID from the EEPROM (words 0x0B, 0x0C).
0	Load Vendor/ Device IDs	1b	This bit, when set to 1b, indicates that the 82576 is to load its PCIe Device IDs from the EEPROM (words 0x0D, 0x11, 0x1D, 0x26).

6.2.3 Subsystem ID (Word 0x0B)

If the *Load Subsystem IDs* in word 0x0A is set, this word is read in to initialize the Subsystem ID. Default value is 0x0. See Section 9.4.14.

6.2.4 Subsystem Vendor ID (Word 0Ch)

If the *Load Subsystem IDs* in word 0x0A is set, this word is read in to initialize the Subsystem Vendor ID. The default value is 0x8086. See Section 9.4.13.

6.2.5 Device ID (Word 0x0D, 0x11)

If the *Load Subsystem IDs* in word 0x0A is set, this word is read in to initialize the Device ID of LAN0, and LAN1 functions, respectively. The default value is 10C9. See Section 9.4.2.

6.2.6 Dummy Device ID (Word 0x1D)

If the *Load Subsystem IDs* in word 0x0A is set, this word is read in to initialize the Device ID of dummy devices. The default value is 0x10A6. See Section 9.4.2.

6.2.7 Initialization Control Word 2 (Word 0x0F)

This is the second word read by the 82576 and contains additional initialization values that:

- Set defaults for some internal registers
- Enable/disable specific features

Bit	Name	Default in EEPROM less mode	Description
15	APM PME# Enable	0b	Initial value of the *Assert PME On APM Wakeup* bit in the Wake Up Control (WUC.APMPME) register. See Section 8.20.1.
14	PCS parallel detect	1b	Enables PCS parallel detect. Mapped to PCS_LCTL.AN TIMEOUT EN bit. See Section 8.18.2
13:12	Pause Capability	11b	Desired pause capability for advertised configuration base page. Mapped to PCS_ANADV.ASM. See Section 8.18.4.
11	ANE	0b	Auto-Negotiation Enable Mapped to PCS_LCTL.AN_ENABLE. See Section 8.18.2.
10:8	FLASH Size Indication	000b	Indicates Flash size according to the following equation: Size = 64 KB * 2**(*Flash Size Indication* field). From 64 KB up to 8 MB in powers of 2. The Flash size impacts the requested memory space for the Flash and Expansion ROM BARs in PCIe configuration space.

7	DMA clock gating enabled	1b	Enables automatic reduction of DMA and MAC frequency. Mapped to STATUS[31]. This bit is relevant only if the L1 indication enable bit is set. See Section 8.2.2.
6	PHY Power Down Enable	1b	When set, enables the PHY to enter a low-power state. This bit is mapped to CTRL_EXT[20]. See Section 8.2.3.
5	Reserved		Reserved - must be zero
4	CCM PLL Shutdown Enable	0b	When set, enables shutting down the CCM PLL in low-power states when the PHY is powered down (such as link disconnect). When cleared, the CCM PLL is not shut down in a low-power state. Reflected in EEDIAG (See Section 8.4.5).
3	L1 indication Enable	0b	When set, enables idle indication to L1 mechanism. See Section 8.2.3.
2	SerDes Low Power Enable	0b	When set, enables the SerDes to enter a low power state when the function is in Dr state. See Chapter 5. and Section 8.2.3.
1	SPD Enable	1b	Smart Power Down – When set, enables PHY Smart Power Down mode. See Section 3.5.7.6.5. This bit is loaded to each of the PHYs, only when the *LAN1_OEM_DIS* and *LAN0_OEM_DIS* bits (word 0x23 bits 8:7) are cleared.
0	LPLU	1b	Low Power Link Up Enables a decrease in link speed in non-D0a states when power policy and power management states dictate it. See Section 3.5.7.6.4. This bit is loaded to each of the PHYs only when *LAN0/1 OEM Bits Disable* (word 0x23 bit 8:7) are cleared.

6.2.8 Software Defined Pins Control - LAN1 (Word 0x10)

This word is used to configure initial settings for the Software Definable Pins (SDPs) for LAN1.

Bit	Name	Default in EEPROM less mode	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. See Section 8.2.3.
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. See Section 8.2.3.
13	PHY_in_LAN_disable	0b	Determines the behavior of the MAC and PHY when a LAN port is disabled through an external pin. 0b = MAC and PHY are kept functional in LAN Disable (to support manageability). 1b = MAC and PHY are powered down in LAN Disable (manageability cannot access the network through this port).
12	Reserved		Reserved - must be zero
11	LAN DISABLE SELECT	0b	LAN Disable When set to 1b, the appropriate LAN is disabled.
10	LAN PCI DISABLE	0b	LAN PCI Disable When set to 1b, the appropriate LAN PCI function is disabled. For example, the LAN is functional for manageability operation but is not connected to the host through the PCIe interface. Reflected in EEDIAG. See Section 8.4.5.
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. SeeSection 8.2.1.
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. SeeSection 8.2.1.
7	SDPVAL[3]	0b	SDP3 Pin – Initial Output Value This bit configures the initial power on value output on SDP3 (when configured as an output) by configuring the initial hardware value of the *SDP3_DATA* bit in the Extended Device Control (CTRL_EXT) register after power up. See Section 8.2.3.

6	SDPVAL[2]	0b	SDP2 Pin – Initial Output Value This bit configures the initial power-on value output on SDP2 (when 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. SeeSection 8.2.3.
5	WD_SDP0	0b	When set, SDP[0] is used as a watchdog timeout indication. When reset, it is used as an SDP (as defined in bits 8 and 0). SeeSection 8.2.1.
4	Giga Disable	0b	When set, GbE operation is disabled. A usage example for this bit is to disable GbE operation if system power limits are exceeded. This bit is loaded to the PHY only when *LAN1_OEM_DIS* (word 0x23 bit 8) is cleared.
3	Disable 1000 in non-D0a	0b	Disables 1000 Mb/s operation in non-D0a states This bit is loaded to the PHY only when *LAN1_OEM_DIS* (word 0x23 bit 8) is cleared. See Section 3.5.7.6.4.
2	D3COLD_WAKEUP_ADVEN	1b	Configures the initial hardware default value of the *ADVD3WUC* bit in the Device Control (CTRL) register following power up. See Section 8.2.1 .
1	SDPVAL[1]	0b	SDP1 Pin – Initial Output Value This bit configures the initial power on value output on SDP1 (when configured as an output) by configuring the initial hardware value of the *SDP1_DATA* bit in the Device Control (CTRL) register after power up. See Section 8.2.1 .
0	SDPVAL[0]	0b	SDP0 Pin – Initial Output Value This bit configures the initial power-on value output on SDP0 (when configured as an output) by configuring the initial hardware value of the *SDP0_DATA* bit in the Device Control (CTRL) register after power up. See Section 8.2.1 .

6.2.9 Software Defined Pins Control - LAN0 (Word 0x20)

This word is used to configure initial settings for the Software Definable Pins (SDPs) for LAN0.

Bit	Name	Default in EEPROM less mode	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. See Section 8.2.3.
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. See Section 8.2.3.
13	PHY_in_LAN_disable	0b	Determines the behavior of the MAC and PHY when a LAN port is disabled through an external pin. 0b = MAC and PHY are kept functional in LAN disable (to support manageability). 1b = MAC and PHY are powered down in LAN disable (manageability cannot access the network through this port). Reflected in EEDIAG. See Section 8.4.5.
12:10	Reserved		Reserved - must be zero
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. See Section 8.2.1.
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. See Section 8.2.1.
7	SDPVAL[3]	0b	SDP3 Pin – Initial Output Value This bit configures the initial power-on value output on SDP3 (when configured as an output) by configuring the initial hardware value of the *SDP3_DATA* bit in the Extended Device Control (CTRL_EXT) register after power up. See Section 8.2.3.

6	SDPVAL[2]	0b	SDP2 Pin – Initial Output Value This bit configures the initial power-on value output on SDP2 (when 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. See Section 8.2.3.
5	WD_SDP0	0b	When set, SDP[0] is used as a watchdog timeout indication. When reset, it is used as an SDP (as defined in bits 8 and 0). See Section 8.2.1.
4	Giga Disable	0b	When set, GbE operation is disabled. A usage example for this bit is to disable GbE operation if system power limits are exceeded. This bit is loaded to the PHY only when *LAN0_OEM_DIS* (word 0x23 bit 7) is cleared.
3	Disable 1000 in non-D0a	0b	Disables 1000 Mb/s operation in non-D0a states. This bit is loaded to the PHY only when *LAN0_OEM_DIS* (word 0x23 bit 7) is cleared. See Section 3.5.7.6.4.
2	D3COLD_WAKEUP_ADVEN	1b	Configures the initial hardware default value of the *ADVD3WUC* bit in the Device Control (CTRL) register following power up. See Section 8.2.1 .
1	SDPVAL[1]	0b	SDP1 Pin – Initial Output Value This bit configures the initial power-on value output on SDP1 (when configured as an output) by configuring the initial hardware value of the *SDP1_DATA* bit in the Device Control (CTRL) register after power up. See See Section 8.2.1 .
0	SDPVAL[0]	0b	SDP0 Pin – Initial Output Value This bit configures the initial power-on value output on SDP0 (when configured as an output) by configuring the initial hardware value of the *SDP0_DATA* bit in the Device Control (CTRL) register after power up. See See Section 8.2.1 .

6.2.10 EEPROM Sizing and Protected Fields (Word 0x12)

Provides indication on EEPROM size and protection.

Note: If the Enable Protection Bit in this word is set and the signature is valid, the software device driver has read but no write access to this word via the EEC and EERD registers; In this case, write access is possible only via an authenticated firmware interface.

Bit	Name	Description
15:14	Signature	The *Signature* field indicates to the 82576 that there is a valid EEPROM present. If the signature field is 01b, EEPROM read is performed, otherwise 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	These bits indicate the EEPROM's actual size. Mapped to EEC[14:11]. 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 = Reserved 1011b = Reserved See Section 8.4.1.

9:5	Reserved	Reserved - must be zero
4	Enable EEPROM Protection	If set, all EEPROM protection schemes are enabled.
3:0	HEPSize	Hidden EEPROM Block Size This field defines the area at the end of the EEPROM memory accessible only by manageability firmware. It can be used to store secured data and other manageability functions. The size in bytes of the secured area equals: 0 bytes if HEPSize equals zero 2^ HEPSize bytes else (for example, 2 B, 4 B, ...32 KB)

6.2.11 Initialization Control 3 (Word 0x14,0x24)

This word controls general initialization values.

Bit	Name	Default in EEPROM less mode	Description
15	SerDes Energy Source	0b	SerDes Energy Source Detection When set to 0b, internal SerDes Rx electrical idle indication. When set to 1b, external LOS signal. This bit also indicates the source of the signal detect while establishing a link in SerDes mode. This bit sets the default value of the *CONNSW.ENRGSRC* bit. See Section 8.2.6.
14	2 wires SFP Enable	0b	2 wires interface SFP Enable 0b = Disabled. When disabled, the 2 wires I/F pads are isolated. 1b = Enabled. Used to set the default value of CTRL_EXT[25]. See Section 8.2.3.
13	LAN Flash Disable	1b	A value of 1b disables the Flash logic. Flash access BAR in the PCI configuration space is disabled.
12:11	Interrupt Pin	00b LAN 0 01b LAN 1	Controls the value advertised in the *Interrupt Pin* field of the PCI Configuration header for this device/function. The encoding of this field is as follow: **Value** **INT Line** ***Interrupt Pin* Field Value** 00b INTA 1 01b INTB 2 10b INTC 3 11b INTD 4 If only a single device/function of the 82576 component is enabled, this value is ignored and the *Interrupt Pin* field of the enabled 82576 reports INTA# usage. See Section 9.4.18.
10	APM Enable	0b	Initial value of *Advanced Power Management Wake Up Enable* bit in the Wake Up Control (WUC.APME) register. Mapped to CTRL[6] and to WUC[0]. See Section 8.2.1 and Section 8.20.1.
9:8	Link Mode	00b	Initial value of *Link Mode* bits of the Extended Device Control (CTRL_EXT.LINK_MODE) register, specifying which link interface and protocol is used by the MAC. 00b = MAC operates with internal copper PHY (1000Base-T). 01b = Reserved. 10b = MAC operates in SGMII mode. 11b = MAC operates in SerDes mode. See Section 8.2.3.
7	LAN Boot Disable	1b	A value of 1b disables the expansion ROM BAR in the PCI configuration space.
6:2	Reserved		Reserved - must be zero.
1	Ext_VLAN	0b	Sets the default for CTRL_EXT[26] bit. Indicates that additional VLAN is expected in this system. See Section 8.2.3.

0	Keep_PHY_Link_Up_En	0b	Enables *No PHY Reset* when the Baseboard Management Controller (BMC) indicates that the PHY should be kept on. When asserted, this bit prevents the PHY reset signal and the power changes reflected to the PHY according to the *MANC.keep_PHY_link_up* value. This bit should be set to the same value at both words (0x14 and 0x24) to reflect the same option to both LANs.

The following tables lists the different combinations of bits 13 and 7:

Flash Disable (Bit 13)	Boot Disable (Bit 7)	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.

PCIe Completion Timeout Configuration (Word 0x15)

Bit	Name	Default in EEPROM less mode	Description
15	Reserved		Reserved - must be zero.
14:12	Reserved		Reserved - must be zero.
11:8	Reserved		Reserved - must be zero
7	Completion Timeout Disable	0b	Disables the PCIe completion timeout mechanism 0b = Completion timeout enabled. 1b = Completion timeout disabled. See Section 8.6.1. This bit is relevant only if the GIO cap field in word 0x1A is set to 01b.
6:5	Completion Timeout Value	00b	Determines the range of the PCIe completion timeout 00b = 50 µs to 10 ms 01b = 10 ms to 250 ms 10b = 250 ms to 4 s 11b = 4 s to 64 s See Section 9.5.5.12. This field is relevant only if the GIO cap field in word 0x1A is set to 01b.
4	Completion Timeout Resend	1b	When set, enables to resend a request once the completion timeout expired 0b = Do not re-send request on completion timeout. 1b = Re-send request on completion timeout. See Section 9.5.5.12
3:0	Reserved	0100b	Reserved

6.2.12 MSI-X Configuration (Word 0x16)

Bit	Name	Default in EEPROM less mode	Description
15:11	MSI_X0_N	0x9	This field specifies the number of entries in MSI-X tables of LAN 0. The range is 0-24. MSI_X_N is equal to the number of entries minus one. See Section 9.5.3.3.
10:6	MSI_X1_N	0x9	This field specifies the number of entries in MSI-X tables of LAN 1. The range is 0-24. MSI_X_N is equal to the number of entries minus one. See Section 9.5.3.3.
5:0	Reserved	0x0	Reserved - must be zero.

6.2.13 PCIe Init Configuration 1 Word (Word 0x18)

This word is used to:

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

Bit	Name	Default in EEPROM less mode	Description
15	Reserved	0b	Reserved - must be zero.
14:12	L1_Act_Ext_Latency	110b (32 µs to 64 µs)	L1 active exit latency for the configuration space. See Section 9.5.5.7.
11:9	L1_Act_Acc_Latency	110b (32 µs to 64 µs)	L1 active acceptable latency for the configuration space. See Section 9.5.5.4.
8:6	L0s_Acc_Latency	011b (512 ns)	L0s acceptable latency for the configuration space. See Section 9.5.5.4.
5:3	L0s_Se_Ext_Latency	110b	L0s exit latency for active state power management (separated reference clock) – (latency between 64 ns – 128 ns). See Section 9.5.5.7.
2:0	L0s_Co_Ext_Latency	101b	L0s exit latency for active state power management (common reference clock) – (latency between 64 ns – 128 ns). See Section 9.5.5.7.

6.2.14 PCIe Init Configuration 2 Word (Word 0x19)

This word is used to set defaults for some internal PCIe configuration registers.

Bit	Name	Default in EEPROM less mode	Description
15	Reserved	0b	Reserved - must be one.
14	IO_Sup	1b	I/O Support (effect I/O BAR request) When set to 1b, I/O is supported.
13	Reserved	0b	Reserved - must be zero.

12	Serial Number enable	0b	When set, the PCIe Serial Number capability is exposed in the configuration space. See Section 9.6.2 for details of the Serial Number capability.
11:8	Reserved		. Reserved - must be 0111b.
7:0	Reserved	0xB0	Reserved - must be 0xB0.

6.2.15 PCIe Init Configuration 3 Word (Word 0x1A)

This word is used to set defaults for some internal registers.

Bit	Name	Default in EEPROM less mode	Description
15:13	Reserved		Reserved - must be zero.
12	Cache_Lsize	0b	Cache Line Size 0b = 64 bytes. 1b = 128 bytes.
11:10	GIO_Cap	10b	PCIe Capability Version The value of this field is reflected in the two LSBs of the capability version in the PCIe CAP register (config space – offset 0xA2). Note that this is not the PCIe version. It is the PCIe capability version. This version is a field in the PCIe capability structure and is not the same as the PCIe version. It changes only when the content of the capability structure changes. For example, PCIe 1.0, 1.0a, and 1.1 all have a capability version of one. PCIe 2.0 has a version two because it added registers to the capabilities structures. See Section 9.5.5.3.
9:8	Max Payload Size	10b	Default packet size 00b = 128 bytes. 01b = 256 bytes. 10b = 512 bytes. 11b = Reserved. See Section 9.5.5.4.
7:6	Lane_Width	10b	Max link width: 00b = 1 lane. 01b = 2 lanes. 10b = 4 lanes. 11b = Reserved. See Section 9.5.5.7.
5:4	Reserved	0	Reserved - must be 10b
3:2	Act_Stat_PM_Sup	11b	Determines support for active state link power management Loaded into the PCIe Active State Link PM Support register. See Section 9.5.5.7.
1	Slot_Clock_Cfg	1b	When set, the 82576 uses the PCIe reference clock supplied on the connector (for add-in solutions).
0	Reserved	0b	Reserved - must be zero.

6.2.16 PCIe Control (Word 0x1B)

This word is used to configure initial settings for the PCIe default functionality.

Bit	Name	Default in EEPROM less mode	Description
15	Reserved	0b	Reserved - must be zero.
14	Dummy Function Enable	0b	0b = When function 0 is disabled, it is replaced by function 1. 1b = When function 0 is disabled, it is replaced with a dummy function.
13:2	Reserved	0x100	Reserved must be 0x100.
1:0	Latency_To_Enter _L1	11b	Period in L0s state before transition into an L1 state: 00b = 64 µsec. 01b = 256 µsec. 10b = 1 msec 11b = 4 msec

6.2.17 LED 1,3 Configuration Defaults (Word 0x1C, 0x2A)

These EEPROM words specify the hardware defaults for the LEDCTL register fields controlling the LED1 (ACTIVITY indication) and LED3 (LINK_1000 indication) output behaviors. Word 0x1C controls LAN0 LEDs behavior and word 0x2A controls LAN1.

Bit	Name	Default in EEPROM less mode	Description
3:0	LED1 Mode	0011b	Initial value of the *LED1_MODE* field specifying what event/state/pattern is displayed on LED1 (ACTIVITY) output. A value of 0011b (0x3) indicates the ACTIVITY state. See Section 8.2.1.
4	Reserved	0b	Reserved - must be zero
5	Reserved	0b	Reserved - must be zero
6	LED1 Invert	0b	Initial value of *LED1_IVRT* field. 0b = Active-low output. See Section 8.2.1.
7	LED1 Blink	1b	Initial value of *LED1_BLINK* field. 0b = Non-blinking. See Section 8.2.1.
15:8	Reserved		Reserved - must be zero
11:8	LED3 Mode	0111b	Initial value of the *LED3_MODE* field specifying what event/state/pattern is displayed on LED3 (LINK_1000) output. A value of 0111b (0x7) indicates 1000 Mb/s operation. See Section 8.2.1.
12	Reserved		Reserved - must be zero.
13	Reserved		Reserved - must be zero
14	LED3 Invert	0b	Initial value of *LED3_IVRT* field. 0b = Active-low output. See Section 8.2.1.
15	LED3 Blink	0b	Initial value of *LED3_BLINK* field. 0b = Non-blinking. See Section 8.2.1.

A value of 0x0703 is used to configure default hardware LED behavior equivalent to previous copper adapters (LED0=LINK_UP, LED1=blinking ACTIVITY, LED2=LINK_100, and LED3=LINK_1000).

6.2.18 Device Rev ID (Word 0x1E)

Bit	Name	Default in EEPROM less mode	Description
15	Power Down Enable	0b	Device Off (dynamic IDDQ) enable/disable bit. See Section 5.2.4.1 for details. Reflected in EEDIAG (See Section 8.4.5).
14	Reserved		Reserved - must be zero.
13	Reserved		Reserved - must be zero
12	LAN 1 iSCSI enable	0b	When set, LAN 1 class code is set to 0x010000 (SCSI) When reset, LAN 1 class code is set to 0x020000 (LAN) See Section 9.4.5
11	LAN 0 iSCSI enable	0b	When set, LAN 0 class code is set to 0x010000 (SCSI) When reset, LAN 0 class code is set to 0x020000 (LAN) See Section 9.4.5
10:8	Reserved		Reserved - must be zero
7:0	DEVREVID	0x1	Device Revision ID. For the 82576 A1, the default value is one. See Section 9.4.5.

6.2.19 LED 0,2 Configuration Defaults (Word 0x1F, 0x2B)

These EEPROM words specify the hardware defaults for the LEDCTL register fields controlling the LED0 (LINK_UP) and LED2 (LINK_100) output behaviors. Word 0x1F controls LAN0 LEDs behavior and word 0x2B controls LAN1.

Bit	Name	Default in EEPROM less mode	Description
3:0	LED0 Mode	0010b	Initial value of the *LED0_MODE* field specifying what event/state/pattern is displayed on LED0 (LINK_UP) output. A value of 0010b (0x2) indicates the LINK_UP state. See Section 8.2.1.
4	Reserved	0b	Reserved - must be zero.
5	Global Blink Mode	0b	Global Blink Mode 0b = Blink at 200 ms on and 200ms off. 1b = Blink at 83 ms on and 83 ms off. See Section 8.2.1.
6	LED0 Invert	0b	Initial value of *LED0_IVRT* field. 0b = Active-low output. See Section 8.2.1.
7	LED0 Blink	0b	Initial value of *LED0_BLINK* field. 0b = Non-blinking. See Section 8.2.1.
11:8	LED2 Mode	0110b	Initial value of the *LED2_MODE* field specifying what event/state/pattern is displayed on LED2 (LINK_100) output. A value of 0110b (0x6) indicates 100 Mb/s operation. See Section 8.2.1.
12	Reserved	0b	Reserved - must be zero.
13	Reserved	0b	Reserved - must be zero
14	LED2 Invert	0b	Initial value of *LED2_IVRT* field. 0b = Active-low output. See Section 8.2.1.
15	LED2 Blink	0b	Initial value of *LED2_BLINK* field. 0b = Non-blinking. See Section 8.2.1.

A value of 0x0602 is used to configure default hardware LED behavior equivalent to previous copper adapters (LED0=LINK_UP, LED1=blinking ACTIVITY, LED2=LINK_100, and LED3=LINK_1000).

6.2.20 Functions Control (Word 0x21)

Bit	Name	Default in EEPROM less mode	Description
15	NC-SI Clock Pad Drive Strength	0b	Defines the drive strength of the NC-SI_CLK_OUT pad. If set, the driving strength is doubled. See Section 11.4.2.4 for details. Reflected in EEDIAG (See Section 8.4.5).
14	NC-SI Data Pad Drive Strength	0b	Defines the drive strength of the NC-SI_DV, NC-SI_RXD[1:0] and NC-SI_ARB_OUT pads. If set, the driving strength is doubled. See Section 11.4.2.4 for details. Reflected in EEDIAG (See Section 8.4.5).
13	NC-SI Output Clock Disable	0b	If set, the clock source is external. In this case, the NC-SI_CLK_OUT pad is kept stable at zero, and the NC-SI_CLK_IN pad is used as an input source of the clock. If cleared, the 82576 outputs the NC-SI clock through the NC-SI_CLK_OUT pad. The NC-SI_CLK_IN pad is still used as an NC-SI clock input. If NC-SI is not used, then this bit should be set. If this bit is cleared, the *Device Power Down Enable* bit in word 0x1E (bit 15) should not be set. Reflected in EEDIAG (See Section 8.4.5).
12	LAN Function Select	0b	When both LAN ports are enabled and *LAN Function Sel* = 0b, LAN 0 is routed to PCI Function 0 and LAN 1 is routed to PCI Function 1. If *LAN Function Sel* = 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]. See See Section 8.6.13 .
11:10	BAR mapping	00b	00b = 32 bit BARS 01b = reserved 10b = 64 bit BARs no I/O BAR 11b = 64 bit BARs no flash BAR. See Section 9.4.11
9	Prefetchable	0	0 = BARs are marked as non prefetchable 1 = BARs are marked as prefetchable See Section 9.4.11
8:6	Reserved		Reserved - must be zero.
5	Reserved		Reserved - must be one.
4:0	Reserved		Reserved - must be zero.

6.2.21 LAN Power Consumption (Word 0x22)

Bit	Name	Default in EEPROM less mode	Description
15:8	LAN D0 Power	0x0	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 100mW units. The power includes also the external logic required for the LAN function. See Section 9.5.1.4.
7:5	Function 0 Common Power	0x0	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 MSBs in the data register that reflects the power values are padded with zeros. See Section 9.5.1.4.
4:0	LAN D3 Power	0x0	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. The power also includes the external logic required for the LAN function. The MSBs in the data register that reflects the power values are padded with zeros. See Section 9.5.1.4.

6.2.22 I/O Virtualization (IOV) Control (Word 0x25)

This word controls IOV functionality.

Bit	Name	Default in EEPROM less mode	Description
15:8	Reserved	0x0	Reserved - must be zero
7:5	Max VFs	0x7	Defines the value of MaxVF exposed in the IOV structure. Valid values are 0-7. The value exposed is the value of this field + one.
4:3	MSI-X table	0x2	Defines the size of the VF function MSI-X table to request. Valid values are 0-2.
2	64-bit Advertisement	1b	0b = VF BARs advertise 32-bit size. 1b = VF BARs advertise 64-bit size.
1	Prefetchable	0b	0b = IOV memory BARS (0 and 3) are declared as non prefetchable. 1b = IOV memory BARS (0 and 3) are declared as prefetchable.
0	IOV Enabled	1b	0b = IOV and ARI capability structures are not exposed as part of the capabilities link list. 1b = IOV and ARI capability structures are exposed as part of the capabilities link list.

6.2.23 IOV Device ID (Word 0x26)

This word defines the device ID for virtual functions.

Bit	Name	Default in EEPROM less mode	Description
15:0	VDev ID	0x10CA	Virtual function device ID.

6.2.24 End of Read-Only (RO) Area (Word 0x2C)

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

Bit	Name	Description
15	Reserved	Reserved - must be zero.
14:0	EORO_area	Defines the end of the area in the EEPROM that is RO. The resolution is one word and can be up to byte address 0xFFFF (0x7FFF words). A value of zero indicates no RO area.

6.2.25 Start of RO Area (Word 0x2D)

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

Bit	Name	Description
15	Reserved	Reserved - must be zero
14:0	SORO_area	Defines the start of the area in the EEPROM that is RO. The resolution is one word and can be up to byte address 0xFFFF (0x7FFF words).

6.2.26 Watchdog Configuration (Word 0x2E)

Bit	Name	Default in EEPROM less mode	Description
15	Watchdog Enable	0b	Enable watchdog interrupt. See Section 8.16.1.
14:11	Watchdog Timeout	0x2	Watchdog timeout period (in seconds). See Section 8.16.1.
10:0	Reserved		Reserved - must be zero

6.2.27 VPD Pointer (Word 0x2F)

This word points to the Vital Product Data (VPD) structure. This structure is available for the NIC vendor to store it's own data. A value of 0xFFFF indicates that the structure is not available.

Bit	Name	Description
15:0	VPD offset	Offset to VPD structure in words. Bit 15 must be set to 0b (VPD area must be in the first 32 Kbytes of the EEPROM).

6.2.28 NC-SI Arbitration enable (Word 0x40)

Bit	Name	Description
15:2	reserved	Reserved - must be 0x0
1	reserved	Reserved - must be 1b
0	NC-SI ARB Enable	0 = NCSI_ARB_IN and NCSI_ARB_OUT pads are not used. NCSI_ARB_IN is pulled up internally to provide stable input. 1 = NCSI_ARB_IN and NCSI_ARB_OUT pads are used.

6.3 Analog Blocks Configuration Structures

6.3.1 Analog configuration pointers start Address (Word 0x17)

Bit(s)	Name	Description
15:0	Address	Defines the word address in the EEPROM of the pointers to the PHY, PCIe, and SerDes initialization spaces.

Note: Word 0x17 points to the pointers of three configuration blocks: SerDes, PHY, and PCIe.

6.3.2 PCIe Initialization Pointer (Offset 0, Relative to Word 0x17 Value)

Bit	Name	Description
15:0	PCIe Config Pointer	Defines the location of the PCIe initialization structure. From this location, the PCIe lane all, PCIe lanes 0/1/2/3, CCM and PLL structures are linked.

6.3.3 PHY Initialization Pointer (Offset 1, Relative to Word 0x17 Value)

Bit	Name	Description
15:0	PHY Config Pointer	Defines the location of the PHY initialization structure. From this location, the PHY structures are linked.

6.3.4 SerDes Initialization Pointer (Offset 2, Relative to Word 0x17 Value)

Bit	Name	Description
15:0	SerDes Config Pointer	Defines the location of the SerDes initialization structure. From this location, the SerDes structures are linked.

6.4 SerDes/PHY/PCIe/PLL/CCM Initialization Structures

6.4.1 Block Header (Offset 0x0)

Bit	Name	Description
15:12	Destination Type	Destination Type. Defines the module type that this block configures: 0x0h = 802.3 PHY. 0x1h = 802.3 SerDes. 0x2h = CCM, GBE PLL. 0x3h = PCIe lane all; write to all four PCIe lanes together. 0x4h = PCIe PLL. 0x5h = PCIe lane 0. 0x6h = PCIe lane 1. 0x7h = PCIe lane 2. 0x8h = PCIe lane 3.
11:10	Next Block	Next Block 00b = The next configuration block proceeds at the end of this one. 01b = This is the last configuration block. 10b = The next configuration block starts at an offset defined by the NBP (second, optional header word). 11b = Reserved.
9:8	Core Destination	Indicates the port to be accessed. 00b = LAN0. 01b = LAN1. 10b = Both cores. This block should be written for both LAN 0 and LAN 1. This field is relevant only if the destination is 802.3 PHY or SerDes blocks.
7:0	Word Count	Size of this structure.

6.4.2 CRC8 (Offset 1)

Bit	Name	Description
15:8	Block CRC	CRC8.
7:0	Reserved	Reserved - must be zero.

6.4.3 Next Buffer Pointer (Offset 2 - Optional)

Bit	Name	Description
15:0	NBP	Pointer to the starting word of the next configuration block.

6.4.4 Address/Data (Offset 3:Word Count)

Bit	Name	Description
15:8	Address	Internal register address that are written to. Refer to the following table.
7:0	Data	Data to write.

ID	Structure Type	Register to Use	Address
0	PHY	MDIC	0x20
1	SerDes	SERDESCTL	0x24
2	CCM	CCMCTL	0x5B48
3	All lanes	GIOANACTLALL	0x5B44
4	PCIe PLL	SCCTL	0x5B4C
5	Lane 0	GIOANACTL0	0x5B34
6	Lane 1	GIOANACTL1	0x5B38
7	Lane 2	GIOANACTL2	0x5B3C
8	Lane 3	GIOANACTL3	0x5B40

For the PHY configuration structure, the description for the configuration words is as follows:

Bit	Name	Description
15:0	MDIC Value	Even words: bits 15:0; Odd words: bits 31:16.

6.5 Firmware Pointers & control words

Words 0x51:0x52 are used to point to the load & no manageability patches and the test structure. Words 0x55:0x57 are used to point to firmware structures specific to PT. Words 0x54 & 0x23 controls some aspects of the FW functionality.

A value of zero for a pointer indicates the relevant structure is not present in the EEPROM.

6.5.1 Loader Patch Pointer (Word 0x51)

Bit	Name	Description
15:0	Pointer	Pointer to loader patch structure. See Section 6.6 for details of the structure.

6.5.2 No Manageability Patch Pointer (Word 0x52)

Bit	Name	Description
15:0	Pointer	Pointer to no manageability patch structure. See Section 6.6 for details of the structure.

6.5.3 PT Patch Configuration Pointer (Word 0x55)

Bit	Name	Description
15:0	Pointer	Pointer to the PT patch configuration pointer structure. See Section 6.6 for details of the structure.

6.5.4 PT LAN0 Configuration Pointer (Word 0x56)

Bit	Name	Description
15:0	Pointer	Pointer to the PT LAN0 configuration pointer structure.See Section 6.7 for details of the structure.

6.5.5 Sideband Configuration Pointer (Word 0x57)

Bit	Name	Description
15:0	Pointer	Pointer to the Sideband configuration pointer structure. See Section 6.8 for details of the structure.

6.5.6 Flex TCO Filter Configuration Pointer (Word 0x58)

Bit	Name	Description
15:0	Pointer	Pointer to the flex TCO configuration pointer structure. See Section 6.9 for details of the structure.

6.5.7 PT LAN1 Configuration Pointer (Word 0x59)

Bit	Name	Description
15:0	Pointer	Pointer to the PT LAN1 configuration pointer structure. See Section 6.7 for details of the structure.

6.5.8 Manageability Capability / Manageability Enable (Word 0x54)

Bit	Name	Description
15	Reserved	Reserved - must be zero.
14	Redirection Sideband Interface	0b = SMBus. 1b = NC-SI.
13:11	Reserved	Reserved - must be zero.
10:8	Manageability Mode	0x0 = None. 0x1 = Reserved. 0x2 = Pass Through (PT) mode. 0x3 = Reserved. 0x4 = Host interface enable only. 0x5:0x7 = Reserved.
7	Port1 Manageability Capable	0 = Not capable 1 = Bits 3 is applicable to port 1.
6	Port0 Manageability Capable	0 = Not capable 1 = Bits 3 is applicable to port 0.
5:4	Reserved	Reserved - must be zero.
3	Pass Through Capable	0b = Disable. 1b = Enable.
2:0	Reserved	Reserved - must be zero.

6.5.9 Management HW Config Control (Word 0x23)

This word contain bits that direct firmware special behavior when configuring the PHY, PCIe, and SerDes interfaces.

Bit	Name	Description
15	LAN1_FTCO_DIS	LAN1 force TCO reset disable (1b disable; 0b enable).
14	LAN0_FTCO_DIS	LAN0 force TCO reset disable (1b disable; 0b enable).
13:10	Reserved	Reserved - must be zero
9	FW Code Exist	If set, indicates to the firmware that there is firmware EEPROM code at address 0x50.
8	LAN1_OEM_DIS	LAN1 OEM bits configuration disable.
7	LAN0_OEM_DIS	LAN0 OEM bits configuration disable.
6	CRC_DIS	PHY, SerDes, and PCIe CRC disable.
5	LAN1_ROM_DIS	LAN1 ROM Disable Disables PHY and SerDes ROM configuration for port 1.
4	LAN0_ROM_DIS	LAN0 ROM Disable Disables PHY and SerDes ROM configuration for port 0.
3	MNG_wake_check_dis	When set, indicates that the firmware to always configure the PHY after power-up without checking if manageability or wake-up is enabled.
2	PCIe ROM Disable	When set, indicates to firmware not to configure the PCIe from the ROM tables.
1	PHY ROM Disable	When set, indicates to firmware not to configure the PHY of both ports from the ROM tables.
0	SerDes ROM Disable	When set, indicates to firmware not to configure the SerDes of both ports from the ROM tables.

6.6 Patch Structure

This structure is used for all the patches in different modes: loader, no manageability, and pass through.

6.6.1 Patch Data Size (Offset 0x0)

Bit	Name	Description
15:0	Data Size (Bytes)	

6.6.2 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	Reserved - must be zero
7:0	CRC8	

6.6.3 Patch Entry Point Pointer Low Word (Offset 0x2)

Bit	Name	Description
15:0	Patch Entry Point Pointer Low Word	

6.6.4 Patch Entry Point Pointer High Word (Offset 0x3)

Bit	Name	Description
15:0	Patch Entry Point Pointer High Word	

6.6.5 Patch Version 1 Word (Offset 0x4)

Bit	Name	Description
15:8	Patch Generation Hour	
7:0	Patch Generation Minutes	

6.6.6 Patch Version 2 Word (Offset 0x5)

Bit	Name	Description
15:8	Patch Generation Month	
7:0	Patch Generation Day	

6.6.7 Patch Version 3 Word (Offset 0x6)

Bit	Name	Description
15:8	Patch Silicon Version Compatibility	0x00 = A0. 0x01 = A1. 0x10 = B0. 0x11 = B1.
7:0	Patch Generation Year	

6.6.8 Patch Version 4 Word (Offset 0x7)

Bit	Name	Description
15:8	Patch Major Number	
7:0	Patch Minor Number	

6.6.9 Patch Data Words (Offset 0x8, Block Length)

Bit	Name	Description
15:0	Patch Firmware Data	

6.7 PT LAN Configuration Structure

Used to pre-configure manageability filters so that pass-thru traffic can be received without explicit configuration by the BMC.

6.7.1 Section Header (Offset 0x0)

Bit	Name	Description
15:8	Block CRC8	
7:0	Block Length	

6.7.2 LAN0 IPv4 Address 0 LSB, MIPAF0 (Offset 0x01)

This value will be stored in the IPV4ADDR0 register (0x58E0).

Bit	Name	Description
15:8	LAN0 IPv4 Address 0 (Byte 1)	Manageability IP Address Filter (Byte 1)
7:0	LAN0 IPv4 Address 0 (Byte 0)	Manageability IP Address Filter (Byte 0)

6.7.3 LAN0 IPv4 Address 0 MSB, MIPAF0 (Offset 0x02)

This value will be stored in the IPV4ADDR0 register (0x58E0).

Bit	Name	Description
15:8	LAN0 IPv4 Address 0 (Byte 3)	Manageability IP Address Filter (Byte 3)
7:0	LAN0 IPv4 Address 0 (Byte 2)	Manageability IP Address Filter (Byte 2)

6.7.4 LAN0 IPv4 Address 1; MIPAF1 (Offset 0x03:0x04)

Same structure as LAN0 IPv4 Address 0.

This value will be stored in the IPV4ADDR1 register (0x58E4).

6.7.5 LAN0 IPv4 Address 2; MIPAF2 (Offset 0x05h:0x06)

Same structure as LAN0 IPv4 Address 0.

This value will be stored in the IPV4ADDR2 register (0x58E8).

6.7.6 LAN0 IPv4 Address 3; MIPAF3 (Offset 0x07h:0x08)

Same structure as LAN0 IPv4 Address 0.

This value will be stored in the IPV4ADDR3 register (0x58EC).

6.7.7 LAN0 MAC Address 0 LSB, MMAL0 (Offset 0x09)

This value will be stored in the MMAL0 register (0x5910).

Bit	Name	Description
15:8	LAN0 MAC Address 0 (Byte 1)	Manageability MAC address low (Byte 1)
7:0	LAN0 MAC Address 0 (Byte 0)	Manageability MAC address low (Byte 0)

6.7.8 LAN0 MAC Address 0 LSB, MMAL0 (Offset 0x0A)

This value will be stored in the MMAL0 register (0x5910).

Bit	Name	Description
15:8	LAN0 MAC Address 0 (Byte 3)	Manageability MAC address low (Byte 3)
7:0	LAN0 MAC Address 0 (Byte 2)	Manageability MAC address low (Byte 2)

6.7.9 LAN0 MAC Address 0 MSB, MMAH0 (Offset 0x0B)

This value will be stored in the MMAH0 register (0x5914).

Bit	Name	Description
15:8	LAN0 MAC Address 0 (Byte 5)	Manageability MAC address high (Byte 5)
7:0	LAN0 MAC Address 0 (Byte 4)	Manageability MAC address high (Byte 4)

6.7.10 LAN0 MAC Address 1; MMAL/H1 (Offset 0x0C:0x0E)

Same structure as LAN0 MAC Address 0.

This value will be stored in the MMAL1/MMAH1 registers (0x5918/1C).

6.7.11 LAN0 MAC Address 2; MMAL/H2 (Offset 0x0F:0x11)

Same structure as LAN0 MAC Address 0.

This value will be stored in the MMAL2/MMAH2 registers (0x5920/24).

6.7.12 LAN0 MAC Address 3; MMAL/H3 (Offset 0x12:0x14)

Same structure as LAN0 MAC Address 0.

This value will be stored in the MMAL3/MMAH3 registers (0x5928/2C).

6.7.13 LAN0 UDP Flex Filter Ports 0:15; MFUTP Registers (Offset 0x15:0x24)

This value will be stored in the MFUTP register (0x5030 - bits 15:0).

Bit	Name	Description
15:0	LAN UDP Flex Filter Value	Management Flex UDP/TCP port

6.7.14 LAN0 VLAN Filter 0:7; MAVTV Registers (Offset 0x25:0x2C)

This value will be stored in the MAVTV[7:0] registers (0x5010 - 0x502C).

Bit	Name	Description
15:12	Reserved	Reserved - must be zero
11:0	LAN0 VLAN Filter Value	VLAN ID value

6.7.15 LAN0 Manageability Filters Valid; MFVAL LSB (Offset 0x2D)

This value will be stored in the MFVAL register (0x5824).

Bit	Name	Description
15:8	VLAN	Indicates whether or not the VLAN filter registers (MAVTV) contain valid VLAN tags. Bit 8 corresponds to filter 0, etc.
7:4	Reserved	Reserved - must be zero.
3:0	MAC	Indicates whether or not the MAC unicast filter registers (MMAH and MMAL) contain valid MAC addresses. Bit 0 corresponds to filter 0, etc.

6.7.16 LAN0 Manageability Filters Valid; MFVAL MSB (Offset 0x2E)

This value will be stored in the MFVAL register (0x5824).

Bit	Name	Description
15:12	Reserved	Reserved - must be zero.
11:8	IPv6	Indicates whether or not the IPv6 address filter registers (MIPAF) contain valid IPv6 addresses. Bit 8 corresponds to address 0, etc. Bit 11 (filter 3) applies only when IPv4 address filters are not enabled (MANC.EN_IPv4_FILTER=0b).
7:4	Reserved	Reserved - must be zero.
3:0	IPv4	Indicates whether or not the IPv4 address filters (MIPAF) contain a valid IPv4 address. These bits apply only when IPv4 address filters are enabled (MANC.EN_IPv4_FILTER=1b)

6.7.17 LAN0 MANC Value LSB (Offset 0x2F)

This value will be stored in the MANC register (0x5820).

Bit	Name	Description
15:0	Reserved	Reserved - must be zero.

6.7.18 LAN0 MANC Value MSB (Offset 0x30)

This value will be stored in the MANC register (0x5820).

Bit	Name	Description
15:11	Reserved	Reserved - must be zero.
10	NET_TYPE	NET TYPE: 0b = pass only un-tagged packets. 1b = pass only VLAN tagged packets. Valid only if FIXED_NET_TYPE is set.
9	FIXED_NET_TYPE	Fixed net type: If set, only packets matching the net type defined by the NET_TYPE field passes to manageability. Otherwise, both tagged and un-tagged packets can be forwarded to the manageability engine.
8	Enable IPv4 Address Filters	When set, the last 128 bits of the MIPAF register are used to store four IPv4 addresses for IPv4 filtering. When cleared, these bits store a single IPv6 filter.
7	Enable Xsum Filtering to MNG	When this bit is set, only packets that pass the L3 and L4 checksum are send to the manageability block.
6	Bypass VLAN	When set, VLAN filtering is bypassed for MNG packets.
5	Enable MNG 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 host memory if they pass the manageability filters.
4:0	Reserved	Reserved - must be zero.

6.7.19 LAN0 Receive Enable 1 (Offset 0x31)

Bit	Name	Description
15:8	Receive Enable Byte 12	BMC SMBus slave address.
7	Enable BMC Dedicated MAC	
6	Reserved	Always set to 1b.
5:4	Notification Method	00b = SMB alert. 01b = Asynchronous notify. 10b = Direct receive. 11b = Reserved.
3	Enable ARP Response	
2	Enable Status Reporting	
1	Enable Receive All	
0	Enable Receive TCO	

6.7.20 LAN0 Receive Enable 2 (Offset 0x32)

Bit	Name	Description
15:8	Receive Enable Byte 14	Alert value.
7:0	Receive Enable Byte 13	Interface value.

6.7.21 LAN0 MANC2H Value LSB (Offset 0x33)

This value will be stored in the MANC2H register (0x5860).

Bit	Name	Description
15:8	Reserved	Must be 0.
7:0	Host Enable	When set, indicates that packets routed by the manageability filters to manageability are also sent to the host. Bit 0 corresponds to decision rule 0, etc.

6.7.22 LAN0 MANC2H Value MSB (Offset 0x34)

This value will be stored in the MANC2H register (0x5860).

Bit	Name	Description
15:0	Reserved	Reserved - must be zero.

6.7.23 Manageability Decision Filters; MDEF0,1 (Offset 0x35)

This value will be stored in the MDEF0 register (0x5890).

Bit	Name	Description
15:12	Flex Port	Controls the inclusion of flex port filtering in the manageability filter decision (OR section). Bit 12 corresponds to flex port 0, etc. (see also bits 11:0 of the next word).
11	Port 0x26F	Controls the inclusion of port 0x26F filtering in the manageability filter decision (OR section).
10	Port 0x298	Controls the inclusion of port 0x298 filtering in the manageability filter decision (OR section).
9	neighbor Discovery	Controls the inclusion of neighbor discovery filtering in the manageability filter decision (OR section).
8	ARP Response	Controls the inclusion of ARP response filtering in the manageability filter decision (OR section).
7	ARP Request	Controls the inclusion of ARP request filtering in the manageability filter decision (OR section).
6	Multicast	Controls the inclusion of Multicast addresses filtering in the manageability filter decision (AND section).
5	Broadcast	Controls the inclusion of broadcast address filtering in the manageability filter decision (OR section).
4	Unicast	Controls the inclusion of unicast address filtering in the manageability filter decision (OR section).
3	IP Address	Controls the inclusion of IP address filtering in the manageability filter decision (AND section).
2	VLAN	Controls the inclusion of VLAN addresses filtering in the manageability filter decision (AND section).
1	Broadcast	Controls the inclusion of broadcast address filtering in the manageability filter decision (AND section).
0	Unicast	Controls the inclusion of unicast address filtering in the manageability filter decision (AND section).

6.7.24 Manageability Decision Filters; MDEF0,2 (Offset 0x36)

This value will be stored in the MDEF0 register (0x5890).

Reserved - must be zero

Bit	Name	Description
15:12	Flex TCO	Controls the inclusion of flex TCO filtering in the manageability filter decision (OR section). Bit 12 corresponds to flex TCO filter 0, etc.
11:0	Flex Port	Controls the inclusion of flex port filtering in the manageability filter decision (OR section). Bit 11 corresponds to flex port 0, etc. (see also bits 15:12 of the previous word).

6.7.25 Manageability Decision Filters; MDEF0,3 (Offset 0x37)

This value will be stored in the MDEF_EXT0 register (0x5930).

Bit	Name	Description
15:12	Reserved	Reserved - must be zero.
11:8	L2 EtherType OR	L2 EtherType - Controls the inclusion of L2 EtherType filtering in the manageability filter decision (OR section).
7:4	Reserved	Reserved - must be zero.
3:0	L2 EtherType AND	L2 EtherType - Controls the inclusion of L2 EtherType filtering in the manageability filter decision (AND section).

6.7.26 Manageability Decision Filters; MDEF0,4 (Offset 0x38)

This value will be stored in the MDEF_EXT0 register (0x5930).

Bit	Name	Description
15:0	Reserved	Reserved - must be zero.

6.7.27 Manageability Decision Filters; MDEF1:6, 1:4 (Offset 0x39:0x50)

Same as words 0x35 to 0x38 for MDEF1:MDEF6.

These values are stored in the MDEF [6:1] registers (0x5894 - 0x58AC) and MDEF_EXT[6:1] registers (0x5934 - 0x594C).

6.7.28 Ethertype filter; METF0, 1 (Offset 0x51)

This value is stored in the METF0 register (0x5060).

Bit	Name	Description
15:0	METF	EtherType value to be compared against the L2 EtherType field in the Rx packet.

6.7.29 Ethertype filter; METF0, 1 (Offset 0x52)

This value is stored in the METF0 register (0x5060).

Bit	Name	Description
15	Reserved	Reserved - must be zero
14	Polarity	0 = Positive filter - forward packets matching this filter to the manageability block. 1 = Negative filter - block packets matching this filter from the manageability block.
13:0	Reserved	Reserved - must be zero

6.7.30 Ethertype filter; METF1:3,1:2 (Offset 0x53:0x58)

Same as words 0x51 and 0x52 for METF1:METF3.

These values are stored in the METF[3:1] registers (0x5064 - 0x506C).

6.7.31 ARP Response IPv4 Address 0 LSB (Offset 0x59)

Bit	Name	Description
15:8	ARP Response IPv4 Address Byte 1	
7:0	ARP Response IPv4 Address Byte 0	

6.7.32 ARP Response IPv4 Address 0 MSB (Offset 0x5A)

Bit	Name	Description
15:8	ARP Response IPv4 Address Byte 3	
7:0	ARP Response IPv4 Address Byte 2	

6.7.33 LAN0 IPv6 Address 0 LSB; MIPAF (Offset 0x5B)

This value will be stored in the MIPAF0 register (0x58B0).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 1	
7:0	LAN0 IPv6 Address 0 Byte 0	

6.7.34 LAN0 IPv6 Address 0 MSB; MIPAF (Offset 0x5C)

This value will be stored in the MIPAF0 register (0x58B0).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 3	
7:0	LAN0 IPv6 Address 0 Byte 2	

6.7.35 LAN0 IPv6 Address 0 LSB; MIPAF (Offset 0x5D)

This value will be stored in the MIPAF1 register (0x58B4).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 5	
7:0	LAN0 IPv6 Address 0 Byte 4	

6.7.36 LAN0 IPv6 Address 0 MSB; MIPAF (Offset 0x5E)

This value will be stored in the MIPAF1 register (0x58B4).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 7	
7:0	LAN0 IPv6 Address 0 Byte 6	

6.7.37 LAN0 IPv6 Address 0 LSB; MIPAF (Offset 0x5F)

This value will be stored in the MIPAF2 register (0x58B8).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 9	
7:0	LAN0 IPv6 Address 0 Byte 8	

6.7.38 LAN0 IPv6 Address 0 MSB; MIPAF (Offset 0x60)

This value will be stored in the MIPAF2 register (0x58B8).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 11	
7:0	LAN0 IPv6 Address 0 Byte 10	

6.7.39 LAN0 IPv6 Address 0 LSB; MIPAF (Offset 0x61)

This value will be stored in the MIPAF3 register (0x58BC).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 13	
7:0	LAN0 IPv6 Address 0 Byte 12	

6.7.40 LAN0 IPv6 Address 0 MSB; MIPAF (Offset 0x62)

This value will be stored in the MIPAF3 register (0x58BC).

Bit	Name	Description
15:8	LAN0 IPv6 Address 0 Byte 15	
7:0	LAN0 IPv6 Address 0 Byte 14	

6.7.41 LAN0 IPv6 Address 1; MIPAF (Offset 0x63:0x6A)

Same structure as LAN0 IPv6 Address 0.

These value are stored in the MIPAF[7:4] registers (0x58C0 - 0x58CC).

6.7.42 LAN0 IPv6 Address 2; MIPAF (Offset 0x6B:0x72)

Same structure as LAN0 IPv6 Address 0.

These value are stored in the MIPAF[11:8] registers (0x58D0 - 0x58DC).

6.8 Sideband Configuration Structure

This section defines paramaters of the SMBus and NC-SI interfaces.

6.8.1 Section Header (Offset 0x0)

Bit	Name	Description
15:8	Block CRC8	
7:0	Block Length	

6.8.2 SMBus Max Fragment Size (Offset 0x1)

Bit	Name	Description
15:0	SMBus Max Fragment Size (Bytes)	

6.8.3 SMBus Notification Timeout and Flags (Offset 0x2)

Bit	Name	Description
15:8	SMBus Notification Timeout (ms)	Timeout until the discarding of a packet not read by the external BMC completes. 0b - No discard.
7:6	SMBus Connection Speed	00b = Slow SMBus connection. 01b = Fast SMBus connection (1 MHz). 10b = Reserved. 11b = Reserved.
5	SMBus Block Read Command	0b = Block read command is C0. 1b = Block read command is D0.
4	SMBus Addressing Mode	0b = Single address mode. 1b = Dual address mode.

3	Reserved	Reserved - must be zero
2	Disable SMBus ARP Functionality	
1	SMBus ARP PEC	
0	Reserved	Reserved - must be zero

6.8.4 SMBus Slave Address (Offset 0x3)

Bit	Name	Description
15:9	SMBus 1 Slave Address	Dual-address mode only.
8	Reserved	Reserved - must be zero.
7:1	SMBus 0 Slave Address	
0	Reserved	Reserved - must be zero.

6.8.5 SMBus Fail-Over Register; Low Word (Offset 0x4)

Bit	Name	Description
15:12	Gratuitous ARP Counter	
11:10	Reserved	Reserved - must be zero.
9	Enable Teaming Fail-over on DX	
8	Remove Promiscuous on DX	
7	Enable MAC Filtering	
6	Enable Repeated Gratuitous ARP	
5	Reserved	Reserved - must be zero.
4	Enable Preferred Primary	
3	Preferred Primary Port	
2	Transmit Pair	
1:0	Reserved	Reserved - set to 1.

6.8.6 SMBus Fail-Over Register; High Word (Offset 0x5)

Bit	Name	Description
15:8	Gratuitous ARP Transmission Interval (seconds)	
7:0	Link Down Fail-over Time	

6.8.7 NC-SI Configuration (Offset 0x6)

Bit	Name	Description
15:11	Reserved	Reserved - must be zero.
10	Reserved.	Reserved - must be zero.

9	NC-SI HW arbitration supported	0b = Not supported. 1b = Supported.
8	NC-SI HW-based packet copy enable	0b = Disable. 1b = Enable.
7:5	Package ID	
4:0	LAN 0 Internal channel ID	Valid values are 0 to 0x1D. Note: If not disabled, LAN1 channel ID is assigned the value of LAN0+1.

6.8.8 NC-SI Hardware arbitration Configuration (Offset 0x8)

Bit	Name	Description
15:0	Token timeout	NC-SI HW-Arbitration TOKEN Timeout (in 16 ns cycles). In order to get the value if NC-SI REF_CLK cycles, this field should be multiplied by 4/5. Setting value to 0 disables the timeout mechanism.

6.8.9 Reserved words (Offset 0x9 - 0xC)

Reserved for future use. Must be zero

6.9 Flex TCO Filter Configuration Structure

Used to pre-configure the manageability-TCO flex filters so that pass-thru traffic can be received without explicit configuration by the BMC. This should be used in configuration with the PT-LAN configuration structure.

6.9.1 Section Header (Offset 0x0)

Bit	Name	Description
15:8	Block CRC8	
7:0	Block Length	

6.9.2 Flex Filter Length and Control (Offset 0x01)

Bit	Name	Description
15:8	Flex Filter Length (Bytes)	
7:5	Reserved	Reserved - must be zero.
4	Last Filter	
3:2	Filter Index (3:0)	
1	Apply Filter to LAN 1	
0	Apply Filter to LAN 0	

6.9.3 Flex Filter Enable Mask (Offset 0x02:0x09)

Bit	Name	Description
15:0	Flex Filter Enable Mask	

6.9.4 Flex Filter Data (Offset 0x0A - Block Length)

Bit	Name	Description
15:0	Flex Filter Data	

6.10 Software Accessed Words

Words 0x03 to 0x07 in the EEPROM image are used for compatibility information. New bits within these fields will be defined as the need arises for determining software compatibility between various hardware revisions.

Words 0x8 and 0x09 are used to indicate the Printed Board Assembly (PBA) number and words 0x42 and 0x43 identifies the EEPROM image.

Words 0x30 to 0x3E have been used for configuration and version values by PXE code. The only exceptions are word 0x3D, which is used for the iSCSI boot configuration and word 0x37 used for alternate MAC address pointer.

6.10.1 Compatibility (Word 0x03)

Bit	Description
15:13	Reserved (set to 000b).
12	ASF SMBus Connected. 0b = Not connected. 1b = Connected.
11	LOM/Not a LOM. 0b = NIC. 1b = LOM.
10	Server/Not a Server NIC. 0b = Client. 1b = Server.
9	Client/Not a Client NIC. 0b = Server. 1b = Client.
8	Retail/OEM. 0b = Retail. 1b = OEM.
7:6	Reserved (set to 00b).
5	Reserved (set to 1b).
4	SMBus Connected. 0b = Not connected. 1b = Connected.

3	Reserved (set to 0b).
2	PCI Bridge/No PCI Bridge. 0b = PCI bridge not present. 1b = PCI bridge present.
1:0	Reserved (set to 00b)

6.10.2 OEM specific (Word 0x04)

Bit	Description
15:12	Control for LED 3. 0001b = Default in STATE1 + Default in STATE2. 0010b = Default in STATE1 + LED is ON in STATE2. 0011b = Default in STATE1 + LED is OFF in STATE2. 0100b = LED is ON in STATE1 + Default in STATE2. 0101b = LED is ON in STATE1 + LED is ON in STATE2. 0110b = LED is ON in STATE1 + LED is OFF in STATE2. 0111b = LED is OFF in STATE1 + Default in STATE2. 1000b = LED is OFF in STATE1 + LED is ON in STATE2. 1001b = LED is OFF in STATE1 + LED is OFF in STATE2.
11:8	Control for LED 2 – same encoding as for LED 3.
7:4	Control for LED 1 – same encoding as for LED 3.
3:0	Control for LED 0 – same encoding as for LED 3.

6.10.3 OEM Specific (Word 0x06, 0x07)

These words are available for OEM use.

6.10.4 EEPROM Image Revision (Word 0x05)

This word is valid only for device starter images and indicates the ID and version of the EEPROM image.

Bit	Description
15:12	EEPROM major version.
11:4	EEPROM minor version.
3:0	EEPROM image ID.

6.10.5 PBA Number (Word 0x08, 0x09)

The nine-digit Printed Board Assembly (PBA) number used for Intel manufactured NICs that are stored in a four-byte field. The dash itself is not stored; neither is the first digit of the 3-digit suffix, as it is always zero for the affected products. Note that through the course of hardware ECOs, the suffix field (byte 4) increments. 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: This PBA number is not related to the MSI-X Pending Bit Array (PBA).

Product	PWA Number	Byte 1	Byte 2	Byte 3	Byte 4
Example	123456-003	12	34	56	03

6.10.6 PXE Main Setup Options (Word 0x30/0x34)

The configuration of the PXE software is controlled by EEPROM on the adapter. The main setup options are stored in word 0x30 and word 0x34. These options are those that can be changed by the user via the Control-S setup menu. The default value for this word is 0x0100. The signature bits in Word 0x31 will be checked by software to determine whether the contents of this word are valid.

Word 0x30 is used to configure the primary port and word 0x34 is used to configure the secondary port.

Bit	Description
15:13	Reserved. Must be 0.
12:10	Bits 12-10 control forcing speed and duplex during driver operation. Valid values are: 000b - Auto negotiate 001b - 10Mbps Half Duplex 010b - 100Mbps Half Duplex 011b - Not valid (treated as 000b) 100b - 10Mbps Full Duplex 101b - 100Mbps Full Duplex 111b - 1000Mbps Full Duplex Only applicable for copper-based adapters. Default value is 000b.
9	Reserved for the 82576 - must be zero
8	Display Setup Message. If the bit is set to 1, the “Press Control-S” message is displayed after the title message. Default value is 1.
7:6	Prompt Time. These bits control how long the “Press Control-S” setup prompt message is displayed, if enabled by bit 8 00 = 2 seconds (default) 01 = 3 seconds 10 = 5 seconds 11 = 0 seconds Note: The Ctrl-S message is not displayed if 0 seconds prompt time is selected.
5	Deprecated. Must be 0.
4:3	Default Boot Selection. These bits select which device is the default boot device. These bits are only used if the agent detects that the BIOS does not support boot order selection or if the MODE field of word 31h is set to MODE_LEGACY. 00 = Network boot, then local boot (default) 01 = Local boot, then network boot 10 = Network boot only 11 = Local boot only
2	Deprecated. Must be 0.
1:0	Protocol Select. These bits select the active boot protocol. 00 = PXE (default value) 01 = Reserved 10 = iSCSI Boot primary port (only if iSCSI Boot is using this adapter) 11 = iSCSI Boot secondary port (only if iSCSI Boot is using this adapter) Only the default value of 00b should be initially programmed into the adapter; other values should only be set by configuration utilities.

6.10.7 PXE Configuration Customization Options (Word 0x31/0x35)

Word 0x31 and word 0x35 of the EEPROM contain settings that can be programmed by an OEM or network administrator to customize the operation of the software. These settings cannot be changed from within the Control-S setup menu. The lower byte contains settings that would typically be configured by a network administrator using an external utility; these settings generally control which setup menu options are changeable. The upper byte is generally settings that would be used by an OEM to control the operation of the agent in a LOM environment, although there is nothing in the agent to prevent their use on a NIC implementation. The default value for this word is 0x4000.

Word 0x31 is used to configure the primary port and word 0x35 is used to configure the secondary port.

Bit	Description
15:14	Signature. Must be set to 01 to indicate that this word has been programmed by the agent or other configuration software.
13:12	Reserved - must be 0
11	Selects Continuous Retry operation. If this bit is set, PCE will NOT transfer control back to the BIOS if it fails to boot due to a network error (such as failure to receive DHCP replies). Instead, it will restart the PXE boot process again. If this bit is set, the only way to cancel PXE boot is for the user to press ESC on the keyboard. Retry will not be attempted due to hardware conditions such as an invalid EEPROM checksum or failing to establish link. Default value is 0.
10:8	Selects the agent's boot order setup mode. This field changes the agent's default behavior in order to make it compatible with systems that do not completely support the BBS and PnP Expansion ROM standards. Valid values and their meanings are: 000b Normal behavior. The agent will attempt to detect BBS and PnP Expansion ROM support as it normally does. (default) 001b Force Legacy mode. The agent will not attempt to detect BBS or PnP Expansion ROM supports in the BIOS and will assume the BIOS is not compliant. The user can change the BIOS boot order in the Setup Menu. 010b Force BBS mode. The agent will assume the BIOS is BBS-compliant, even though it may not be detected as such by the agent's detection code. The user can NOT change the BIOS boot order in the Setup Menu. 011b Force PnP Int18 mode. The agent will assume the BIOS allows boot order setup for PnP Expansion ROMs and will hook interrupt 18h (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. The user can NOT change the BIOS boot order in the Setup Menu. 100b Force PnP Int19 mode. The agent will assume the BIOS allows boot order setup for PnP Expansion ROMs and will hook interrupt 19h (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. The user can NOT change the BIOS boot order in the Setup Menu. 101b - 111b Reserved for future use. If specified, is treated as a value of 000b.
7:6	Reserved - must be 0
5	Disable Flash Update. If this bit is set to 1, the user is not allowed to update the flash image using DIMX. Default value is 0
4	Disable Legacy Wakeup Support. If this bit is set to 1, the user is not allowed to change the Legacy OS Wakeup Support menu option. Default value is 0.
3	Disable Boot Selection. If this bit is set to 1, the user is not allowed to change the boot order menu option. Default value is 0.
2	Disable Protocol Select. If set to 1, the user is not allowed to change the boot protocol. Default value is 0.
1	Disable Title Message. If this bit is set to 1, the title message displaying the version of the Boot Agent is suppressed; the Control-S message is also suppressed. This is for OEMs who do not wish the boot agent to display any messages at system boot. Default value is 0.
0	Disable Setup Menu. If this bit is set to 1, the user is not allowed to invoke the setup menu by pressing Control-S. In this case, the EEPROM may only be changed via an external program. Default value is 0.

6.10.8 PXE Boot Agent Version Number (Word 0x32)

Word 32h of the EEPROM is used to store the version of the boot agent that is stored in the flash image. Some older diagnostic tools read this word in order to report the version of the Boot Agent in the flash.

Bit	Description
15:12	Boot agent major version
11:8	Boot agent minor version
7:0	Boot agent build number

6.10.9 Flash Capabilities (Word 0x33)

Word 33h of the EEPROM is used to enumerate the boot technologies that have been programmed into the flash. This is generally updated by flash configuration tools and is not updated by the option ROM code.

Bit	Description
15:14	Signature. Must be set to 01 to indicate that this word has been programmed by the agent or other configuration software.
13:5	Reserved (set to 0)
4	iSCSI Boot is present in flash if set to 1
3	EFI UNDI driver is present in flash if set to 1.
2	RPL module is present in flash if set to 1.
1	PXE UNDI driver is present in flash if set to 1.
0	PXE Base Code is present in flash if set to 1.

6.10.10 PXE Reserved words (Words 0x36, 0x38 - 0x3C, 0x3E)

These words are reserved for other PXE usage.

6.10.11 iSCSI Boot Configuration Start Address (Word 0x3D)

Bit	Name	Description
15:0	Address	Defines the word address in the EEPROM space where the iSCSI boot configuration structure starts.

6.10.12 iSCSI Module Structure

Configuration Item	Max Size in Bytes	Comments
iSCSI Boot Signature	2	‘i’, ‘S’
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	Version of this structure. Should be set to one.
Checksum	1	Holds 8-bit checksum of this structure.
Flags	2	0x01 to enable iSCSI boot. 0x02 to valid configuration. This should be set to one if the configuration information in this structure is valid, otherwise zero. 0x03 to enable DHCP. 0x04:0x05 to configuration prompt: 0 = 0 Seconds. 1 = 2 Seconds. 2 = 3 Seconds. 3 = 5 Seconds. 0x06:ox0A to # of connection retries.

Initiator IP	4	DHCP flag not set. This field should contain the configured IP address. DHCP flag set. This field should be set to zero or save the last configured IP address.
Initiator Subnet Mask	4	DHCP flag not set. This field should contain the configured subnet mask. DHCP flag set. This field should be set to zero or save the last configured subnet mask.
Target IP	4	IP address of iSCSI target.
Target Port	2	IP port of iSCSI target with a default of 3260.
CHAP Password	32	
CHAP User Name	255 + 1	Variable length field.
Initiator Name	255 + 1	Variable length field.

The maximum amount of boot configuration information that is 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 of 128. The following field begins at 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.

6.10.13 Alternate MAC address pointer (Word 0x37)

This word may point to a location in the EEPROM containing additional MAC addresses used by system management functions. If the additional MAC addresses are not supported, the word shall be set to 0xFFFF

6.10.14 Checksum Word (Word 0x3F)

The checksum word (0x3F) is used to ensure that the base EEPROM image is a valid image. The value of this word should be calculated such that after adding all the words (0x00:0x3F), including the checksum word itself, the sum should be 0xBABA. The initial value in the 16-bit summing register should be 0x0000 and the carry bit should be ignored after each addition.

Note: Hardware does not calculate the word 0x3F checksum 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 0x00:0x3F during EEPROM reads in order to determine validity of the EEPROM image; this field is provided strictly for software verification of EEPROM validity. All hardware configurations based on word 0x00:0x3F content is based on the validity of the *Signature* field of EEPROM Initialization Control Word 1 (*Signature* must be 01b).

6.10.15 Image Unique ID (Word 0x42, 0x43)

These words contain a unique 32-bit ID for each image generated by Intel to enable tracking of images and comparison to the original image if testing a customer EEPROM image.

§ §

NOTE: ***This page intentionally left blank.***

7. Inline Functions

7.1 Receive Functionality

7.1.1 Rx Queues Assignment

A received packet goes through three stages of filtering as shown in Figure 7-1. Figure 7-1 describes a switch-like structure that is used in virtualization mode to route packets between the network port (top of drawing) and one or more virtual ports (bottom of figure), where each virtual port can be associated with a virtual machine, an IOVM, a VMM, or the like.

The first step in queue assignment is to make sure that the packet is received by the port. This is done by a set of L2 filters as described in Section 7.1.2.

The second stage is specific to virtualization environments and defines the virtual ports (called pools) that are the targets for the Rx packet. A packet can be associated with any number of ports/pools and the selection process as described in Section 7.1.1.2.

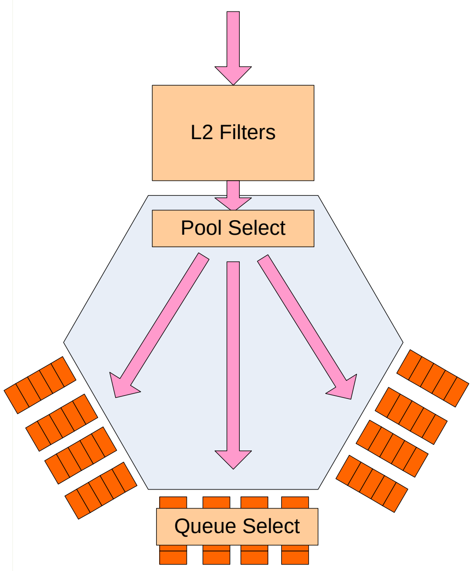


Figure 7-1. Stages in Packet Filtering

In the third stage, a receive packet that successfully passed the Rx filters is associated with one of more receive descriptor queues as described in this section.

The following filter mechanisms determine the destination of a receive packet. These are described briefly in this section and in full details in separate sections:

- Virtualization — In a virtualized environment, DMA resources are shared between more than one software entity (operating system and/or software device driver). This is done by allocating receive descriptor queues to virtual partitions (VMM, IOVM, VMs, or VFs). Allocating queues to virtual partitions is done in sets, each with the same number of queues called queue pools or pools. Virtualization assigns to each received packet one or more pool indices. Packets are routed to a pool based on their pool index and other considerations, such Receive Side Scaling (RSS). See Section 7.1.1.2 for details on routing for virtualization.
- RSS — RSS distributes packet processing between several processor cores by assigning packets into different descriptor queues. RSS assigns to each received packet an RSS index. Packets are routed to one of a set of Rx queues based on their RSS index and other considerations such as virtualization. See Section 7.1.1.7 for details on RSS.
- L2 Ethertype filters — These filters identify packets by their L2 Ether type and assign them to receive queues. Examples of possible uses are LLDP packets and 802.1X packets. See Section 7.1.1.4 for mode details. The 82576 incorporates four Ether-type filters.
- L3/L4 5-tuple filters — These filters identify specific L3/L4 flows or sets of L3/L4 flows. Each filter consists of a 5-tuple (protocol, source and destination IP addresses, source, and destination TCP/UDP port) and routes packets into one of the Rx queues. The 82576 has eight such filters. See Section 7.1.1.5 for details.
- TCP SYN filters — The 82576 might route TCP packets with their *SYN* flag set into a separate queue. *SYN* packets are often used in *SYN* attacks to load the system with numerous requests for new connections. By filtering such packets to a separate queue, security software can monitor and act on *SYN* attacks. See Section 7.1.1.6 for mode details.

Typically, 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 one of the 16 receive queues in host memory, and updating the state of a receive descriptor.

Note: Maximum supported received-packet size is 9.5 KB (9728 bytes).

A received packet is allocated to a queue based on the previous criteria and the following order:

- Queue by L2 Ether-type filters (if a match)
- If RFCTL.SYNQFP is 0b, then:
  - — Queue by L3/L4 5-tuple filters (if a match)
  - — Queue by SYN filter (if a match)
- If RFCTL.SYNQFP is 1b, then:
  - — Queue by SYN filter (if a match)
  - — Queue by L3/L4 5-tuple filters (if a match)
- Define a pool (in case of virtualization)
- Queue by RSS.

Table 7-1lists the allocation of the queues in each of the modes.

Table 7-1. Queue Allocation[1]

Virtualization	RSS	Queue allocation
Disabled	Disabled	One default queue (MRQC.DEF_Q)
	Enabled	Up to 16 queues by RSS.
Enabled	Disabled	One queue per VM (queues 0-7 for VM 0-7).
	Enabled	Two queues per VM (queues 0, 8; 1, 9; 2, 10; 3, 11; 4, 12; 5, 13; 6, 14; 7; 15 for VM 0-7, respectively). Spread between the queues by RSS.

1. On top of this allocation, the special filters can override the queueing decision.

7.1.1.1 Queuing in a Non-Virtualized Environment

A received packet is assigned to a queue in the following manner:

- L2 Ether-type filters **—** Each filter identifies one of 16 Rx queues.
- SYN filter **—** Identifies one of 16 Rx queues.
- L3/L4 5-tuple filters **—** Each filter identifies one of 16 Rx queues.
- RSS filters - Identifies one of 2 x 8 queues through the RSS index. The following modes are supported:
  - — No RSS **—** The default queue as defined in MRQC.DEF_Q is used for packets that do not meet any of the previous conditions.
  - — RSS **—** A set of 16 queues is allocated for RSS. The queue is identified through the RSS index. Note that it is possible to use a subset of the 16 queues.

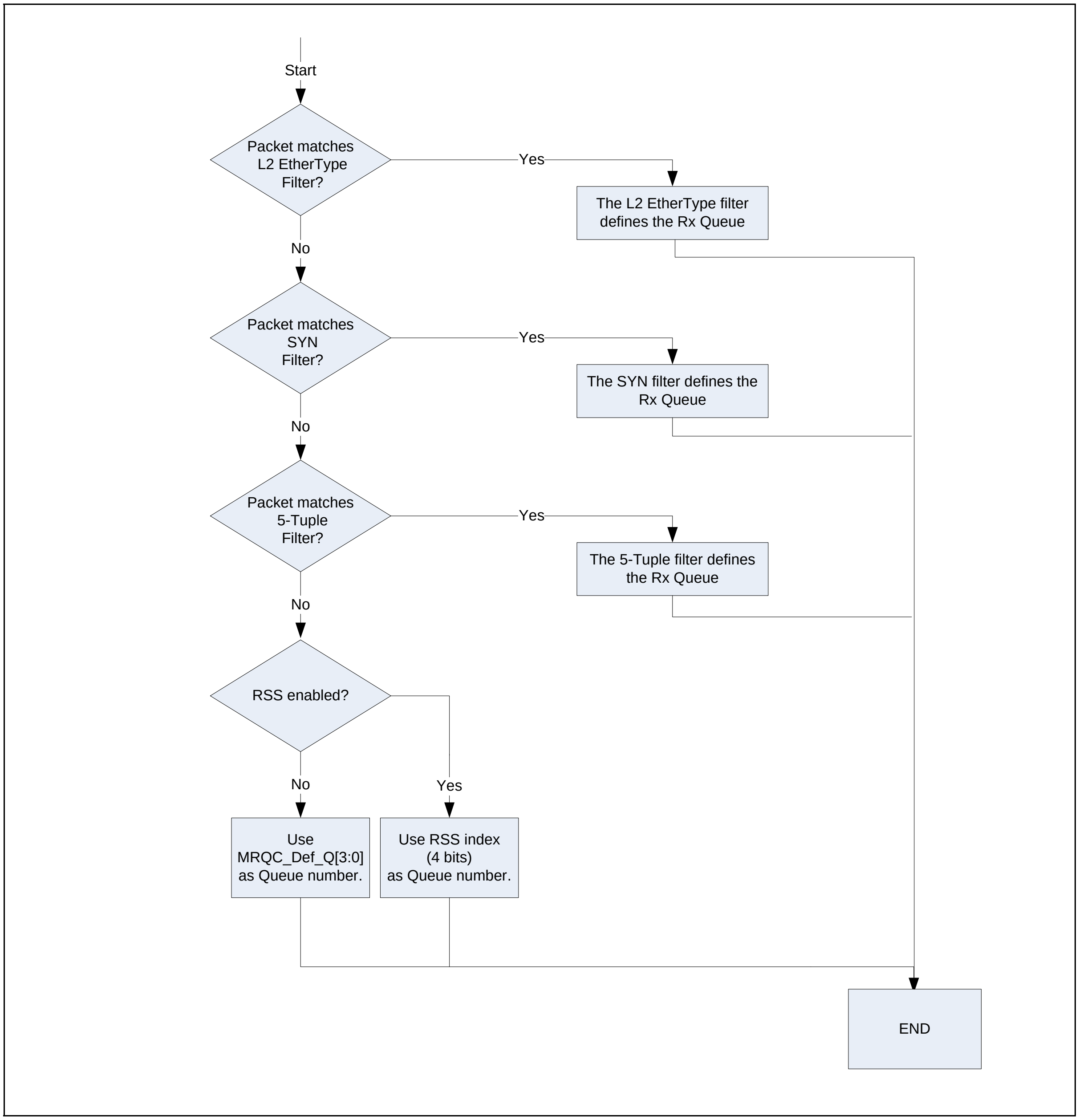

Start
Packet matches L2 EtherType Filter?
Yes
The L2 EtherType filter defines the Rx Queue
No
Packet matches SYN Filter?
Yes
The SYN filter defines the Rx Queue
No
Packet matches 5-Tuple Filter?
Yes
The 5-Tuple filter defines the Rx Queue
No
RSS enabled?
No
Yes
Use MRQC_Def_Q[3:0] as Queue number.
Use RSS index (4 bits) as Queue number.
END

Figure 7-2. Rx Queuing Flow (Non-Virtualized)

7.1.1.2 Rx Queuing in a Virtualized Environment

The 16 Rx queues are allocated to a pre-configured number of queue sets called pools. In Next Generation VMDq mode, system software allocates the pools to the VMM, an IOVM, or to VMs. In IOV mode, each pool is associated with a VF.

Incoming packets are associated with pools based on their L2 characteristics as described in Section 7.9.3. This section describes the following stage, where an Rx queue is assigned to each replication of the Rx packet as determined by its pool's association.

A received packet is assigned to a queue within a pool in the following manner:

- L2 Ether-type filters — Each filter identifies a specific queue, belonging to some pool (the queue designation determines the pool and is usually allocated to the VMM or a service operating system).
- SYN filter — Not supported in VT modes.
- L3/L4 5-tuple filters — Each filter is associated with a single Rx queue, belonging to a specific pool.
- RSS filters — The following modes are supported:
  - No RSS — A single queue is allocated per pool (queue 0 of each pool).
  - RSS — All 16 queues are allocated to pools. Note that it is possible to enable RSS usage per pool using the *VMOLR.RSSE* bit. If the packet is not suitable for RSS, then a queue 0 for each pool is used.

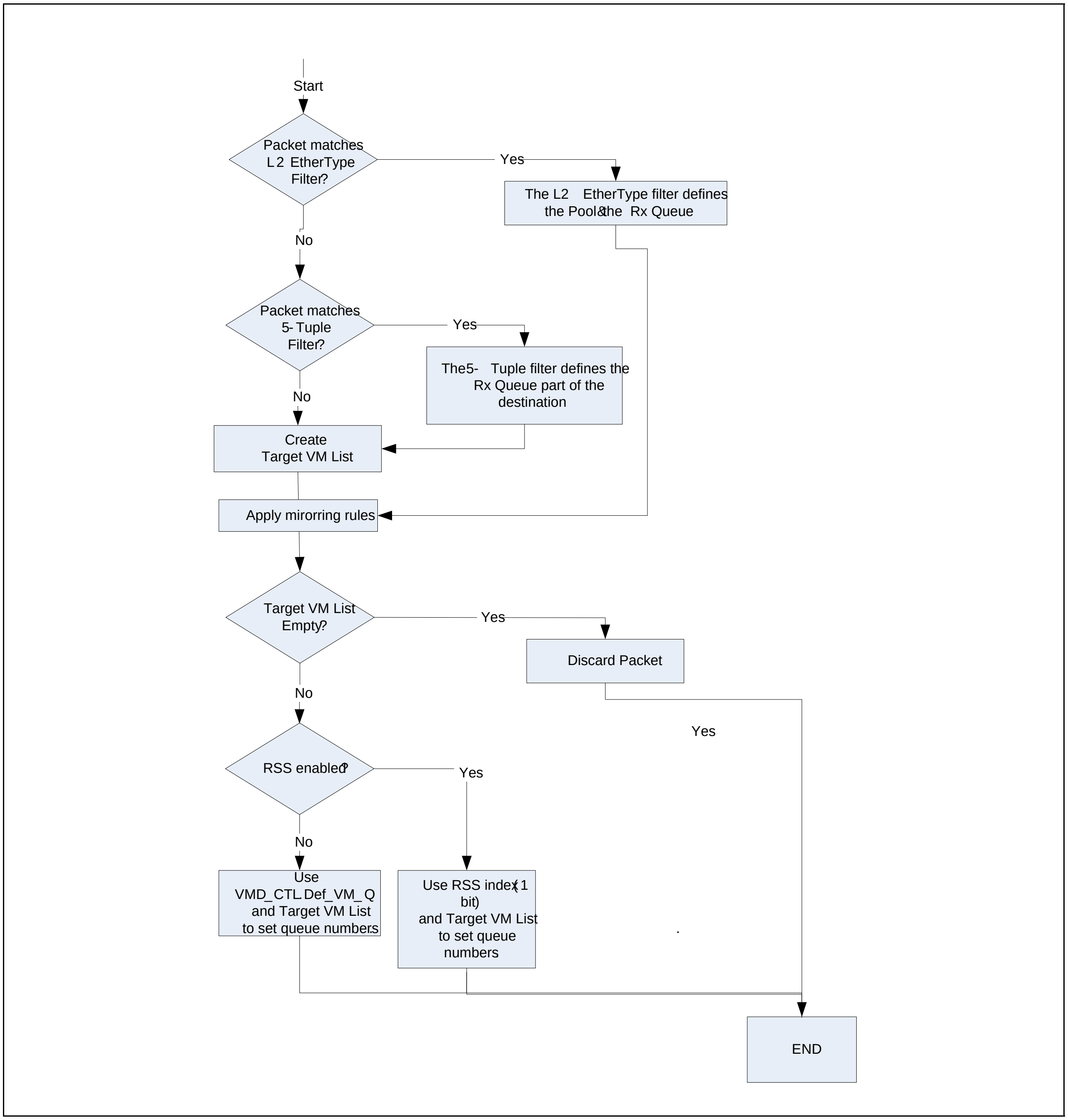

Start
Packet matches L2 EtherType Filter?
Yes
The L2 EtherType filter defines the Pool&the Rx Queue
No
Packet matches 5-Tuple Filter?
Yes
The5- Tuple filter defines the Rx Queue part of the destination
No
Create Target VM List
Apply mirorring rules
Target VM List Empty?
Yes
Discard Packet
No
Yes
RSS enabled?
Yes
No
Use VMD_CTL.Def_VM_Q and Target VM List to set queue numbers
Use RSS index(1 bit) and Target VM List to set queue numbers
END

Figure 7-3. Rx Queuing Flow (Virtualization)

7.1.1.3 Queue Configuration Registers

Configuration registers (CSRs) that control queue operation are replicated per queue (total of 16 copies of each register). Each of the replicated registers correspond to a queue such that the queue index equals the serial number of the register (such as register 0 corresponds to queue 0, etc.). Registers included in this category are:

- RDBAL and RDBAH — Rx Descriptor Base
- RDLEN — RX Descriptor Length
- RDH — RX Descriptor Head
- RDT — RX Descriptor Tail
- RXDCTL — Receive Descriptor Control
- RXCTL — Rx DCA Control

CSRs that define the functionality of descriptor queues are replicated per VF index to allow for a separate configuration in a virtualization environment (total of eight copies of each register). Each of the replicated registers correspond to a set of queues with the same VF index, such that the VF index of the queue identifies the serial number of the register. Registers included in this category are:

- SRRCTL — Split and Replication Receive Control
- PSRTYPE — Packet Split Receive type

7.1.1.4 L2 Ether-Type Filters

These filters identify packets by L2 Ether-type and assign them to a receive queue. The following usages have been identified:

- IEEE 802.1X packets — Extensible Authentication Protocol over LAN (EAPOL).
- Time sync packets (such as IEEE 1588) — Identifies Sync or Delay_Req packets

The 82576 incorporates eight Ether-type filters.

The *Packet Type* field in the Rx descriptor captures the filter number that matched with the L2 Ether-type. See Section 7.1.5 for decoding of the *Packet Type* field.

The Ether-type filters are configured via the ETQF register as follows:

- The *EType* field contains the 16-bit Ether-type compared against all L2 type fields in the Rx packet.
- The *Filter Enable* bit enables identification of Rx packets by Ether-type according to this filter. If this bit is cleared, the filter is ignored for all purposes.
- The *Rx Queue* field contains the absolute destination queue for the packet.
- The *1588 Time Stamp* field indicates that the packet should be time stamped according to the IEEE 1588 specification.
- The *Queue Enable* field enables forwarding Rx packets based on the Ether-type defined in this register.

Special considerations for Virtualization modes:

- Packets that match an Ether-type filter are diverted from their original pool (the pool identified by the L2 filters) to the pool used as the pool to which the queue in the *Queue* field belongs. In other words, The L2 filters are ignored in determining the pool for such packets.
- The same applies for multi-cast packets. A single copy is posted to the pool defined by the filter.
- Mirroring rules:
  - — If a pool is being mirrored, the pool to which the queue in the *Queue* field belongs is used to determine if a packet that matches the filter should be mirrored.

— The queue inside the pool (indicated by the *Queue* field) is used for both the original pool and the mirroring pool.

7.1.1.5 L3/L4 5-Tuple Filters

These filters identify specific L3/L4 flows or sets of L3/L4 flows. Each filter consists of a 5-tuple (protocol, source and destination IP addresses, source and destination TCP/UDP port) and forwards packets into one of the Rx queues. In a virtualized environment, each filter can be associated with one specific VF and a packet must match the L2 conditions for that VF to match the 5-tuple filter.

The 82576 incorporates eight such filters.

The 5-tuple filters are configured via the FTQF, SPQF, IMIR, IMIR_EXT, DAQF & SAQF registers as follows (per filter):

- Protocol — Identifies the IP protocol, part of the 5-tuple queue filters. Enabled by a bit in the *Mask* field.
- Source address — Identifies the IP source address, part of the 5-tuple queue filters. Enabled by a bit in the *Mask* field. Only IPv4 addresses are supported.
- Destination address — Identifies the IP destination address, part of the 5-tuple queue filters. Enabled by a bit in the *Mask* field. Only IPv4 addresses are supported.
- Source port — Identifies the TCP/UDP source port, part of the 5-tuple queue filters. Enabled by a bit in the *Mask* field.
- Destination port — Identifies the TCP/UDP destination port, part of the 5-tuple queue filters. Enabled if the IMIR.PORT_BP field is cleared.
- Size threshold — Identifies the length of the packet that should trigger the filter. This is the length as received by the host, not including any part of the packet removed by hardware. Enabled by the *Size_BP* field.
- Control Bits — Identify TCP flags that might be part of the filtering process. Enabled by the CtrlBit_BP field.
- Rx queue — Determines the Rx queue for packets that match this filter. Only the LSB bits are used:
  - — In a non-virtualized configuration, the *Rx Queue* field contains the queue serial number.
  - — In the virtualized configuration, the *Rx Queue* field contains the queue serial number within the set of queues of the VF associated (via the *VF* field) with this filter. In this case, the packet is sent to all VFs in the VF index list (see Section 7.1.1.2 for details) in the queue defined in the filter.
- Queue enable — Enables forwarding a packet that uses this filter.
- VF — Identifies the VF associated with this filter by its VF index (virtualization modes only). A packet must match the VF filters (such as MAC address) and the 5-tuple filter for this filter to apply.

Note: The above field should not be set to match a mirror port (such as a port that receives promiscuous traffic), as it influences the queuing of packets sent to mirrored port.

- VF Mask — Determines if the *VF* field participates in the 5-tuple match or is ignored:
  - — Must be set to 1b in non-virtualized case
  - — In a virtualized configuration:
    - When set to 0b, only unicast packets that match the *VF* field are candidates for this filter.
    - When set to 1b, unicast, multicast, and broadcast packets might all match with the 5-tuple filter. VF association is not checked. The *Rx Queue* field defines a queue for each VF.
- Mask — A5-bit field that masks each of the fields in the 5-tuple (L4 protocol, IP addresses, TCP/UDP ports). The filter is a logical AND of the non-masked 5-tuple fields. If all 5-tuple fields are masked, the filter is not used for queue forwarding.

Note: If more than one 5-tuple filter with the same priority are matched by the packet, the first filter (lowest ordinal number) is used in order to define the queue destination of this packet.

The immediate interrupt and 1588 actions are defined by the OR of all the matching filters.

Filtering rules for IPv6 packets are:

- If a filter defines at least one of the IP source and destination addresses, then an IPv6 packet always misses such a filter.
- If a filter masks both the IP source and destination addresses, then an IPv6 packet is compared against the remaining fields of the filter.
- Tunnelled packets are not matched by the 5-tuple filters.

Note: These filters are not available for VM to VM traffic forwarding.

7.1.1.6 SYN Packet Filters

The 82576 might forward TCP packets whose *SYN* flag is set into a separate queue. SYN packets are often used in SYN attacks to load the system with numerous requests for new connections. By filtering such packets to a separate queue, security software can monitor and act on SYN attacks.

SYN filters are configured via the SYNQF registers as follows:

- Queue En — Enables forwarding of SYN packets to a specific queue.
- Rx Queue field — Contains the destination queue for the packet.

This filter is not to be used in a virtualized environment.

7.1.1.7 Receive-Side Scaling (RSS)

RSS is a mechanism to distribute received packets into several descriptor queues. Software then assigns each queue to a different processor, sharing the load of packet processing among several processors.

As described in Section 7.1.1.7, the 82576 uses RSS as one ingredient in its packet assignment policy (the others are the various filters and virtualization). The RSS output is a RSS index. The 82576’s global assignment uses these bits (or only some of the LSBs) as part of the queue number.

RSS is enabled in the MRQC register. The RSS *Status* field in the descriptor write-back is enabled when the *RXCSUM.PCSD* bit is set (fragment checksum is disabled). RSS is therefore mutually exclusive with UDP fragmentation. Also, support for RSS is not provided when legacy receive descriptor format is used.

When RSS is enabled, the 82576 provides software with the following information as required by Microsoft* RSS or for device driver assistance:

- A Dword result of the Microsoft* RSS hash function, to be used by the stack for flow classification, is written into the receive packet descriptor (required by Microsoft* RSS).
- A 4-bit RSS *Type* field conveys the hash function used for the specific packet (required by Microsoft* RSS).

Figure 7-4 shows the process of computing an RSS output:

1. The receive packet is parsed into the header fields used by the hash operation (such as IP addresses, TCP port, etc.).
2. A hash calculation is performed. The 82576 supports a single hash function, as defined by Microsoft* RSS. The 82576 does not indicate to the software device driver which hash function is used. The 32-bit result is fed into the packet receive descriptor.

3. The seven LSBs of the hash result are used as an index into a 128-entry indirection table. Each entry provides a 3-bit RSS output index.

When RSS is disabled, packets are assigned an RSS output index = zero. System software might enable or disable RSS at any time. While disabled, system software might update the contents of any of the RSS-related registers.

When multiple requests queues are enabled in RSS mode, un-decodable packets are assigned an RSS output index = zero. The 32-bit tag (normally a result of the hash function) equals zero.

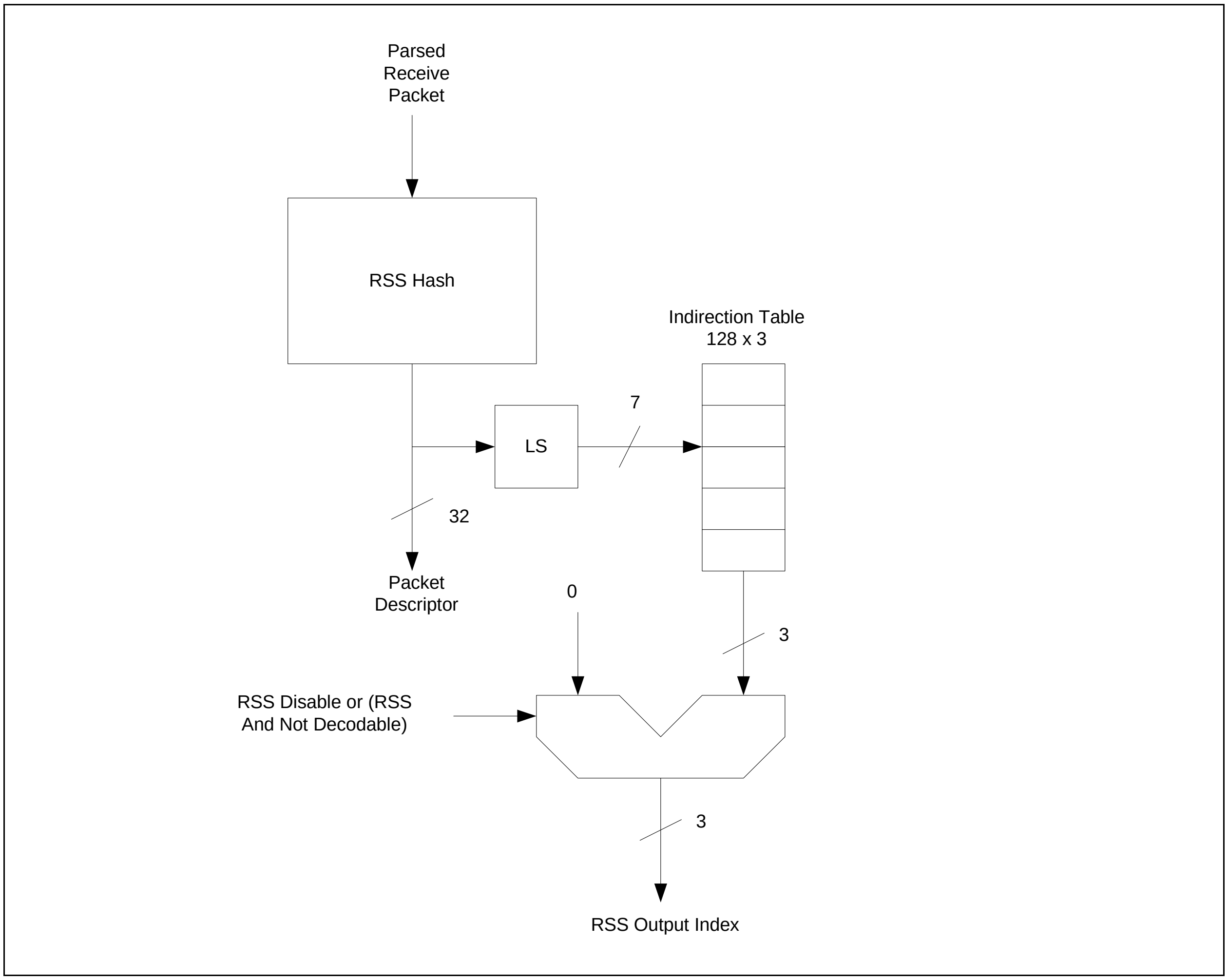


Figure 7-4. RSS Block Diagram

7.1.1.7.1 RSS Hash Function

Section 7.1.1.7.1 provides a verification suite used to validate that the hash function is computed according to Microsoft* nomenclature.

The 82576 hash function follows Microsoft* definition. A single hash function is defined with several variations for the following cases:

- TcpIPv4 — The 82576 parses the packet to identify an IPv4 packet containing a TCP segment per the criteria described later in this section. If the packet is not an IPv4 packet containing a TCP segment, RSS is not done for the packet.

- IPv4 — The 82576 parses the packet to identify an IPv4 packet. If the packet is not an IPv4 packet, RSS is not done for the packet.
- TcpIPv6 — The 82576 parses the packet to identify an IPv6 packet containing a TCP segment per the criteria described later in this section. If the packet is not an IPv6 packet containing a TCP segment, RSS is not done for the packet.
- TcpIPv6Ex — The 82576 parses the packet to identify an IPv6 packet containing a TCP segment with extensions per the criteria described later in this section. If the packet is not an IPv6 packet containing a TCP segment, RSS is not done for the 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).
- IPv6Ex — The 82576 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. In this case, the IPv6 hash is used. If the packet is not an IPv6 packet, RSS is not done for the packet.
- IPv6 — The 82576 parses the packet to identify an IPv6 packet. If the packet is not an IPv6 packet, receive-side-scaling is not done for the packet.

The following additional cases are not part of the Microsoft* RSS specification:

- UdpIPV4 — The 82576 parses the packet to identify a packet with UDP over IPv4.
- UdpIPV6 — The 82576 parses the packet to identify a packet with UDP over IPv6.
- UdpIPV6Ex — The 82576 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 (such as IP options can be parsed, packet not encrypted).
- The packet is not 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 can 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:

1. Try using the TcpIPv4 function.
2. Try using IPV4_UDP function.
3. Try using the IPv4 function.

IPv6 packet:

1. If TcpIPv6Ex is enabled, try using the TcpIPv6Ex function; else if TcpIPv6 is enabled try using the TcpIPv6 function.
2. If UdpIPv6Ex is enabled, try using UdpIPv6Ex function; else if UpdIPv6 is enabled try using UdpIPv6 function.
3. 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 previously mentioned rules, it is assigned an RSS output index = zero. The 32-bit tag (normally a result of the hash function) equals zero.

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 hash functions:

- Ordering is little endian in both bytes and bits. For example, the IP address 161.142.100.80 translates into 0xa18e6450 in the signature.
- A “^ “denotes bit-wise 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, it is considered that 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 length of 320 bits (40 bytes); the key is typically supplied through the RSS Random Key Register (RSSRK).

The algorithm works by examining each bit of the hash input from left to right. Intel’s nomenclature defines left and right for a byte-array as follows: Given an array K with k bytes, Intel’s nomenclature assumes that the array is laid out as shown:

```
K[0] K[1] K[2] … K[k-1]
```

K[0] is the left-most byte, and the MSB of K[0] is the left-most bit. K[k-1] is the right-most byte, and the LSB 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 an without a TCP header.

7.1.1.7.1.1 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);
```

7.1.1.7.1.2 Hash for IPv4 with UDP

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);
```

7.1.1.7.1.3 Hash for IPv4 without TCP

Concatenate SourceAddress and DestinationAddress into one single byte-array

```
Input[8] = @12-15, @16-19
Result = ComputeHash(Input, 8)
```

7.1.1.7.1.4 Hash for IPv6 with TCP

Similar to above:

```
Input[36] = @8-23, @24-39, @40-41, @42-43
Result = ComputeHash(Input, 36)
```

7.1.1.7.1.5 Hash for IPv6 with UDP

Similar to above:

```
Input[36] = @8-23, @24-39, @40-41, @42-43
Result = ComputeHash(Input, 36)
```

7.1.1.7.1.6 Hash for IPv6 without TCP

```
Input[32] = @8-23, @24-39
Result = ComputeHash(Input, 32)
```

7.1.1.7.2 Indirection Table

The indirection table is a 128-entry structure, indexed by the seven LSBs of the hash function output. Each entry of the table contains the following:

- Bits [3:0] - RSS index

Note: In RSS mode, all bits are used. In Next Generation VMDq + RSS mode only bit 0 is used.

System software might update the indirection 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 takes effect on a specific packet boundary.

7.1.1.7.3 RSS Verification Suite

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
```

7.1.1.7.3.1 IPv4

Table 7-2. IPv4

Destination Address/Port	Source Address/Port	IPv4 Only	IPv4 With TCP
161.142.100.80:1766	66.9.149.187:2794	0x323e8fc2	0x51ccc178
65.69.140.83:4739	199.92.111.2:14230	0xd718262a	0xc626b0ea

Table 7-2. IPv4

12.22.207.184:38024	24.19.198.95:12898	0xd2d0a5de	0x5c2b394a
209.142.163.6:2217	38.27.205.30:48228	0x82989176	0xafc7327f
202.188.127.2:1303	153.39.163.191:44251	0x5d1809c5	0x10e828a2

7.1.1.7.3.2 IPv6

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

Table 7-3. IPv6

Destination Address/Port	Source Address/Port	IPv6 Only	IPv6 With TCP
3ffe:2501:200:1fff::7 (1766)	3ffe:2501:200:3::1 (2794)	0x2cc18cd5	0x40207d3d
ff02::1 (4739)	3ffe:501:8::260:97ff:fe40:efab (14230)	0x0f0c461c	0xdde51bbf
fe80::200:f8ff:fe21:67cf (38024)	3ffe:1900:4545:3:200:f8ff:fe21:67cf (44251)	0x4b61e985	0x02d1feef

7.1.1.7.4 Association Through MAC Address

Each of the 24 MAC address filters can be associated with a VF/VM. The *VIND* field in the Receive Address High (RAH) register determines the target VM. Packets that do not match any of the MAC filters (such as promiscuous) are assigned with the default VT.

Software can program different values to the MAC filters (any bits in RAH or RAL) at any time. The 82576 would respond to the change on a packet boundary but does not guarantee the change to take place at some precise time.

7.1.2 L2 Packet Filtering

The receive packet filtering role is to determine which of the incoming packets are allowed to pass to the local system and which of the incoming packets should be dropped since they are not targeted to the local system. Received packets can be destined to the host, to a manageability controller (BMC), or to both. This section describes how host filtering is done, and the interaction with management filtering.

As shown in Figure 7-5, host filtering has three stages:

1. Packets are filtered by L2 filters (MAC address, unicast/multicast/broadcast). See Section 7.1.2.1 for details.
2. Packets are then filtered by VLAN if a VLAN tag is present. See Section 7.1.2.2 for details.
3. Packets are filtered by the manageability filters (port, IP, flex, other). See Section 10.4 for details.

A packet is not forwarded to the host if any of the following takes place:

1. The packet does not pass MAC address filters as described later in this section.

- The packet does not pass VLAN filtering as described later in this section.
- The packet passes manageability filtering and then the manageability filters determine that the packet should not pass to host as well (see Section 10.4 and the MANC2H register).

A packet that passes receive filtering as previously described might still be dropped due to other reasons. Normally, only good packets are received. These are defined as those packets with no Under Size Error, Over Size Error, Packet Error, Length Error and CRC Error are detected. However, if the *store-bad-packet* bit is set (*FCTRL.SBP*), then bad packets that don't pass the filter function is 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 and the *store-bad-packet* bit.

If there is insufficient space in the receive FIFO, hardware drops the packet and indicates the missed packet in the appropriate statistics registers.

Note: CRC errors before the SFD are ignored. Any packet must have a valid SFD in order to be recognized by the 82576 (even bad packets).

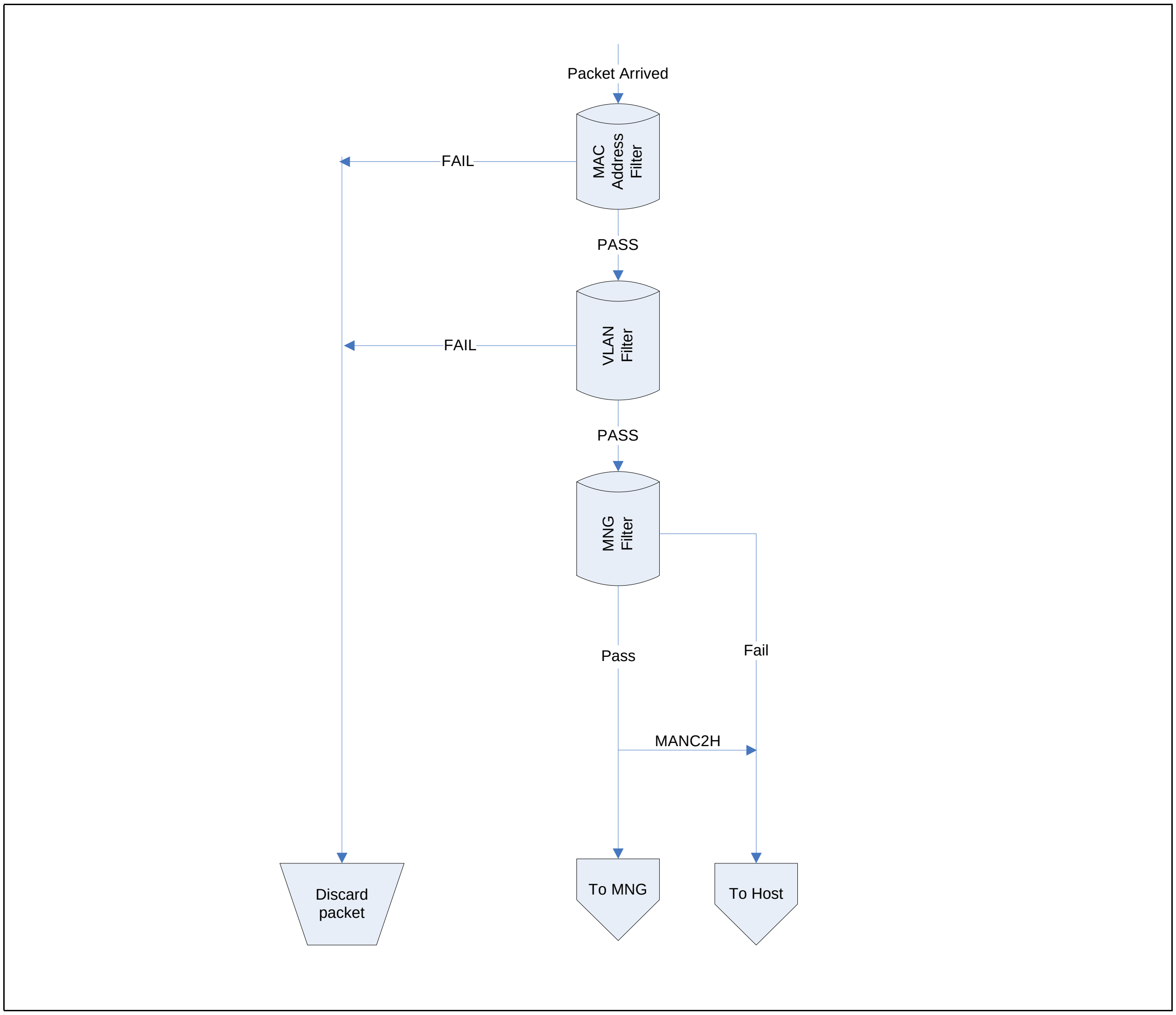


Figure 7-5. Rx Filtering Flow Chart

7.1.2.1 MAC Address Filtering

Figure 7-6 shows the MAC address filtering. A packet passes successfully through the MAC address filtering if any of the following conditions are met:

1. It is a unicast packet and promiscuous unicast filtering is enabled.
2. It is a multicast packet and promiscuous multicast filtering is enabled.
3. It is a unicast packet and it matches one of the unicast MAC filters (host or manageability).
4. It is a multicast packet and it matches one of the multicast filters.
5. It is a broadcast packet and Broadcast Accept Mode (BAM) is enabled. Note that in this case, for manageability traffic, the packet does not go through VLAN filtering (VLAN filtering is assumed to match).

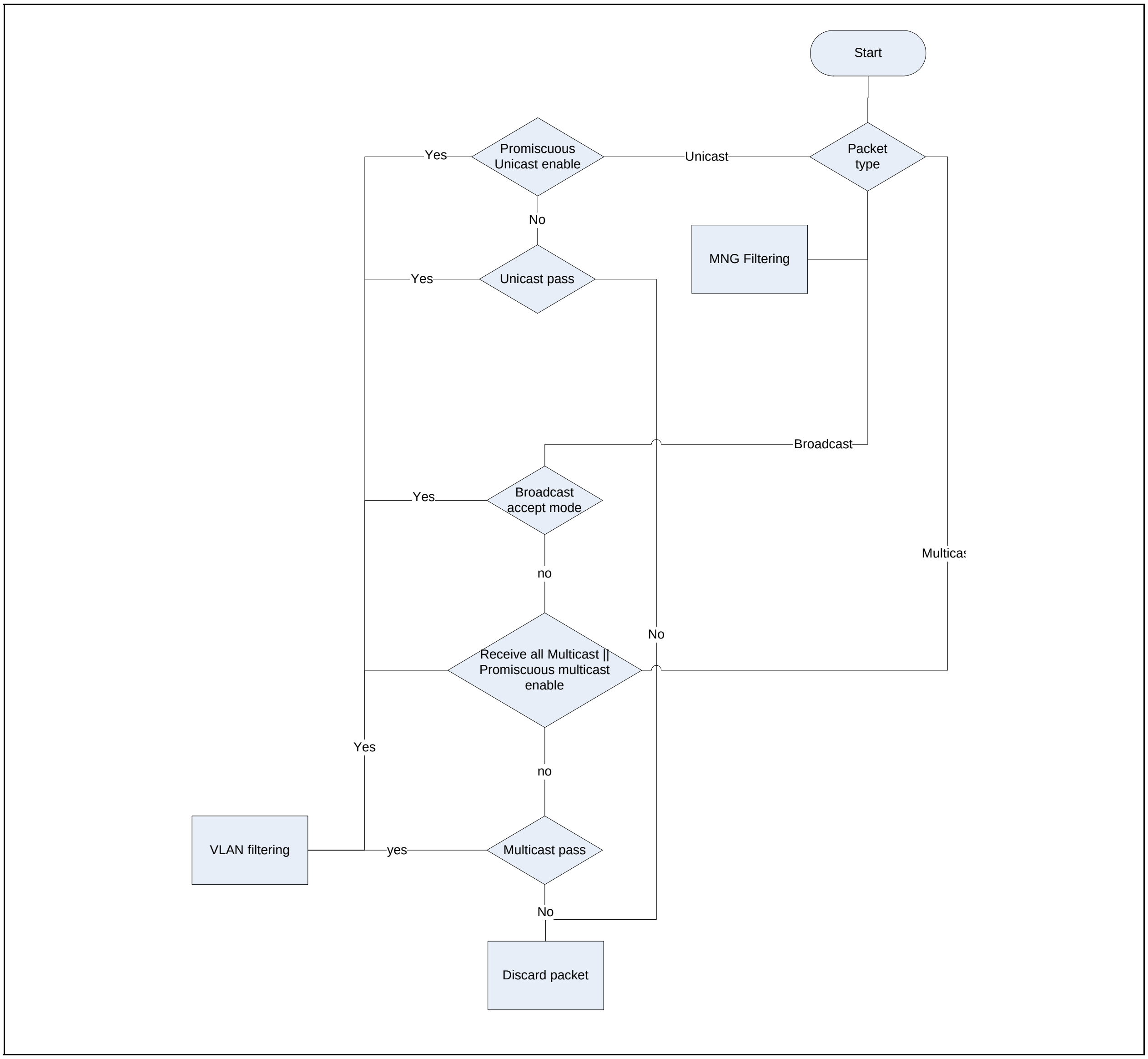


Figure 7-6. MAC Address Rx Filtering Flow Chart

7.1.2.1.1 Unicast Filter

The entire MAC address is checked against the 16 host unicast addresses and four management unicast addresses (if enabled). The 16 host unicast addresses are controlled by the host interface (the BMC must not change them). The other four addresses are dedicated to management functions and are only accessed by the BMC. The destination address of incoming packet must exactly match one of the pre-configured host address filters or the manageability address filters. These addresses can be unicast or multicast. Those filters are configured through RAL, RAH, MMAL, and MMAH registers.

Promiscuous Unicast — Receive all unicasts. Promiscuous unicast mode can be set/cleared only through the host interface (not by the BMC) and it is usually used when the 82576 is used as a sniffer.

Unicast Hash Table — Destination address matching the Unicast Hash Table (UTA).

7.1.2.1.2 Multicast Filter (Partial)

The 12-bit portion of incoming packet multicast address must exactly match Multicast Filter Address (MFA) in order to pass multicast filtering. Those 12 bits out of 48 bits of the destination address can be selected by the *MO* field of RCTL (Section 8.10.1). These entries can be configured only by the host interface and cannot be controlled by the BMC. Packets received according to this mode have the *PIF* bit in the descriptor set to indicate imperfect filtering that should be validated by the software device driver.

Promiscuous Multicast — Receive all multicast. Promiscuous multicast mode can be set/cleared only through the host interface (not by the BMC) and it is usually used when the 82576 is used as a sniffer.

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

7.1.2.2 VLAN Filtering

A receive packet that successfully passed MAC address filtering is then subjected to VLAN header filtering.

1. If the packet does not have a VLAN header, it passes to the next filtering stage.

Note: If extended VLAN is enabled (*CTRL_EXT.EXTENDED_VLAN* is set), it is assumed that the first VLAN tag is an extended VLAN and it is skipped. All next stages refer to the second VLAN.

2. If the packet has a VLAN header and it passes a valid manageability VLAN filter, then is passes to the next filtering stage.
3. If VLAN filtering is disabled (*RCTL.VFE* bit is cleared), the packet is forwarded to the next filtering stage.
4. If the packet has a VLAN header, and it matches an enabled host VLAN filter, the packet is forwarded to the next filtering stage.
5. If the packet has a VLAN header and MANC.Bypass VLAN is set, the packet is forwarded to the next filtering stage, but is candidate for manageability forwarding only.
6. Otherwise, the packet is dropped.

Figure 7-7 shows the VLAN filtering flow.

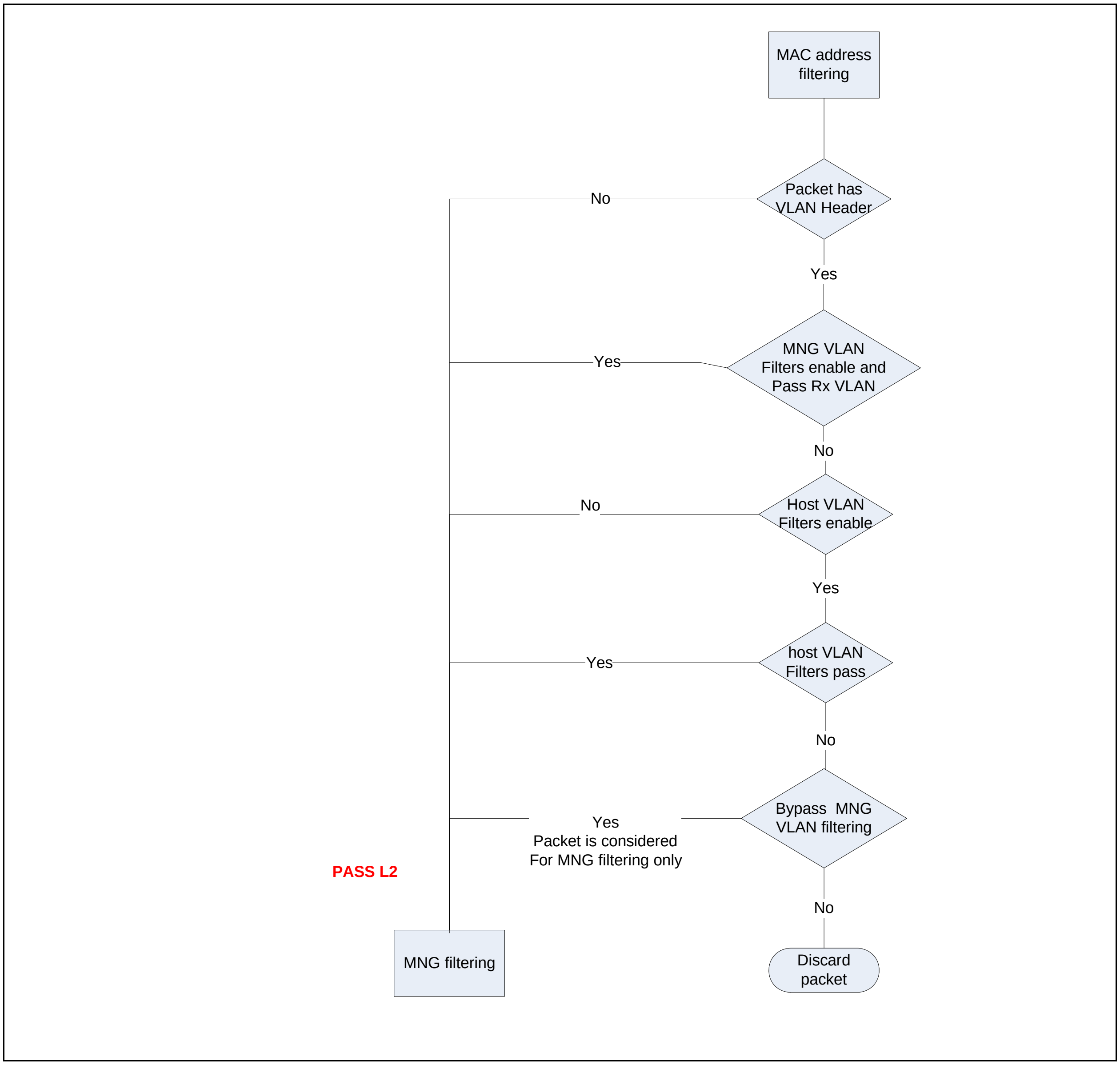


Figure 7-7. VLAN Filtering

7.1.2.3 Manageability Filtering

Manageability filtering is described in Section 10.4.

Figure 7-8 shows the manageability portion of the packet filtering and it is brought here to make the receive packet filtering functionality description complete.

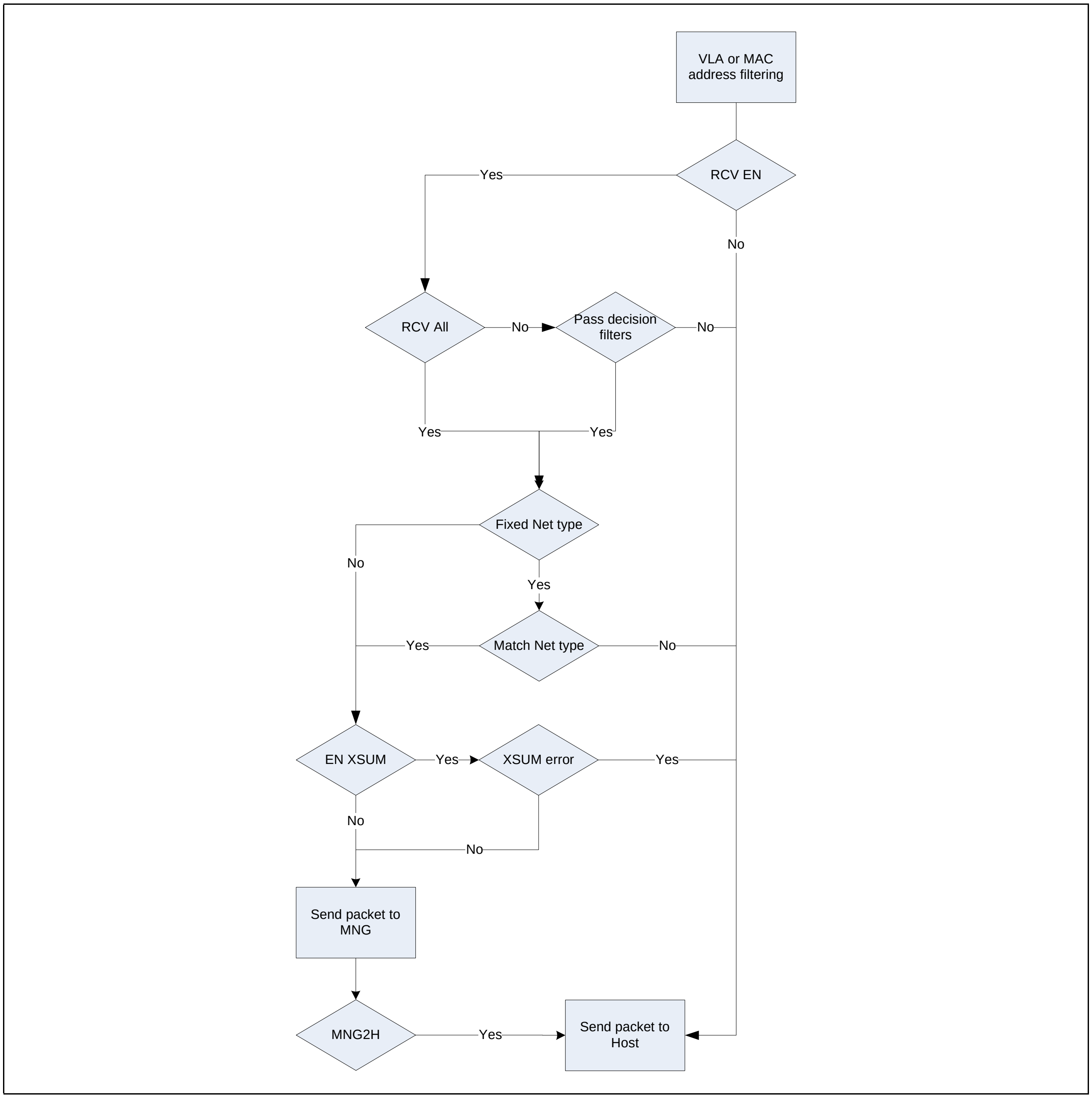


Figure 7-8. Manageability Filtering

Note: The manageability engine might decide to snoop or redirect part of the received packets, according to the external BMC instructions and the EEPROM settings.

7.1.3 Receive Data Storage

7.1.3.1 Host Buffers

Each descriptor points to a one or more memory buffers that are designated by the software device driver to store packet data.

The size of the buffer can be set using either the generic *RCTL.BSIZE* field, or the per queue *SRRCTL[n].BSIZEPACKET* field.

The receive buffer size is selected by bit settings in the Receive Control (RCTL.BSIZE). The register supports buffer sizes of 256, 512, 1024, and 2048 bytes. See section 12.7.1 for details.

If *SRRCTL[n].BSIZEPACKET* is set to zero for any queue, the buffer size defined by *RCTL.BSIZE* is used. Otherwise, the buffer size defined by *SRRCTL[n].BSIZEPACKET* is used.

In addition, for advanced descriptor usage the *SRRCTL.BSIZEHEADER* field is used to define the size of the buffers allocated to headers.

The 82576 places no alignment restrictions on receive memory 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 might have no knowledge of a specific device's buffer alignment requirements.

Note: When the *No-Snoop Enable* bit is used in advanced descriptors, the buffer address is 16-bit (2-byte) aligned.

7.1.3.2 On-Chip Rx Buffers

The 82576 contains a 64 KBytes packet buffer that can be used to store packets until they are forwarded to the host.

In addition, to support the forwarding of local packets as described in Section 7.9.3, a switch buffer of 20 KBytes is provided. This buffer serves as a receive buffer for all the local traffic.

7.1.3.3 On-Chip descriptor Buffers

The 82576 contains a 32 descriptor cache for each receive queue used to reduce the latency of packet processing and to optimize the usage of PCIe bandwidth by fetching and writing back descriptors in bursts. The fetch and writeback algorithm are described in Section 7.1.6 and Section 7.1.7.

7.1.4 Legacy 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 SRRCTL[n],DESCTYPE = 000b, the 82576 uses the legacy Rx descriptor as shown in Table 7-4. The shaded areas indicate fields that are modified by hardware upon packet reception (so-called descriptor write-back).

Note: Legacy descriptors should not be used when advanced features such as Virtualization or security features are activated.

Table 7-4. Legacy Receive Descriptor (RDESC) Layout

	63 48	47 40	39 32	31 16	15 0
0	Buffer Address [63:0]				
8	VLAN Tag	Errors	Status	Fragment Checksum	Length

After receiving a packet for the 82576, hardware stores the packet data into the indicated buffer and writes the length, packet checksum, status, errors, and status fields.

Packet Buffer Address (64) - Physical address of the packet buffer.

Length Field (16)

Length covers the data written to a receive buffer including CRC bytes (if any). Software must read multiple descriptors to determine the complete length for a packet that spans multiple receive buffers.

Fragment Checksum (16)

This field is used to provide the fragment checksum value. This field equals to the unadjusted 16-bit ones complement of the packet. Checksum calculation starts at the L4 layer (after the IP header) until the end of the packet excluding the CRC bytes. In order to use the fragment checksum assist to offload L4 checksum verification, software might need to back out some of the bytes in the packet. For more details see Section 7.1.10.2

Status Field (8)

Status information indicates whether the descriptor has been used and whether the referenced buffer is the last one for the packet. See Table 7-5 for the layout of the *Status* field. Error status information is shown in Figure 7-9.

Table 7-5. Receive Status (RDESC.STATUS) Layout

7	6	5	4	3	2	1	0
PIF	IPCS	L4CS	UDPCS	VP	Rsv	EOP	DD

- PIF (bit 7) - Passed in-exact filter
- IPCS (bit 6) - Ipv4 checksum calculated on packet
- L4CS (bit 5) - L4 (UDP or TCP) checksum calculated on packet
- UDPCS (bit 4) - UDP checksum calculated on packet
- VP (bit 3) - Packet is 802.1q (matched VET); indicates strip VLAN in 802.1q packet
- RSV (bit 2) - Reserved
- EOP (bit 1) - End of packet
- DD (bit 0) - Descriptor done

EOP and DD

The following table lists the meaning of these bits:

Table 7-6. Receive Status Bits

DD	EOP	Description
0b	0b	Software setting of the descriptor when it hands it off to the hardware.
0b	1b	Reserved (invalid option).
1b	0b	A completion status indication for a non-last descriptor of a packet that spans across multiple descriptors. In a single packet case, DD indicates that the hardware is done with the descriptor and its buffers. Only the *Length* fields are valid on this descriptor.
1b	1b	A completion status indication of the entire packet. Note that software Might take ownership of its descriptors. All fields in the descriptor are valid (reported by the hardware).

VP Field

The *VP* field indicates whether the incoming packet's type matches VET. For example, if the packet is a VLAN (802.1q) type, it is set if the packet type matches VET and CTRL.VME is set. It also indicates that VLAN has been stripped in 802.1q packet. For more details, see Section 7.4.

IPCS (Ipv4 Checksum), L4CS (L4 Checksum), and UDPCS (UDP Checksum)

The meaning of these bits is shown in the table below:

Table 7-7. IPCS, L4CS, and UDPCS

L4CS	UDPCS	IPCS	Functionality
0b	0b	0b	Hardware does not provide checksum offload. Special case: Hardware does not provide UDP checksum offload for IPV4 packet with UDP checksum = 0b
1b	0b	1b / 0b	Hardware provides IPv4 checksum offload if IPCS is active and TCP checksum is offload. A pass/fail indication is provided in the *Error* field – IPE and L4E.
0b	1b	1b / 0b	Hardware provides IPpv4 checksum offload if IPCS is active and UDP checksum is offload. A pass/fail indication is provided in the *Error* field – IPE and L4E.

Refer to Table 7-18 for a description of supported packet types for receive checksum offloading. Unsupported packet types do not have the *IPCS* or *L4CS* bits set. IPv6 packets do not have the *IPCS* bit set, but might have the *L4CS* bit set if the 82576 recognized the TCP or UDP packet.

PIF

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. Multicast packets passing only the Multicast Vector (MTA) or unicast packets passing only the Unicast Hash Table (UTA) but not any of the MAC address exact filters (RAH, RAL) have *PIF* set. In addition, the following condition causes *PIF* to be cleared:

- The DA of the packet is a multicast address and promiscuous multicast is set (RCTL.MPE = 1b).
- The DA of the packet is a broadcast address and accept broadcast mode is set (RCTL.BAM = 1b)

A MAC control frame forwarded to the host (RCTL.PMCF = 0b) that does not match any of the exact filters, has the *PIF* bit set.

Error Field (8)

Most error information appears only when the *store-bad-packet* bit (RCTL.SBP) is set and a bad packet is received. See Figure 7-9 for a definition of the possible errors and their bit positions.

Table 7-8. RXE, LPE,

7	6	5	4	3	2	1	0
RXE	IPE	L4E	Reserved				

- RXE (bit 7) - RX Data Error
- IPE (bit 6) - Ipv4 Checksum Error
- L4E (bit 5) - TCP/UDP Checksum Error
- Reserved (bit 4:0)

IPE/L4E

The IP and TCP checksum error bits from Figure 7-9 are valid only when the IPv4 or TCP/UDP checksum(s) is performed on the received packet as indicated via IPCS and L4CS. These, along with the other error bits, are valid only when the *EOP* and *DD* bits are set in the descriptor.

Note: Receive checksum errors have no affect on packet filtering.

If receive checksum offloading is disabled (RXCSUM.IPOFL and RXCSUM.TUOFL), the *IPE* and *L4E* bits are 0b.

RXE

The RXE error bit is asserted in one of two cases (software might distinguish between these errors by monitoring the respective statistics registers):

1. CRC error is detected. CRC can be a result of reception of /V/ symbol on the TBI interface (see section 3.5.3.3.2) or assertion of RxERR on the MII/GMII interface or bad EOP or lose of sync during packet reception. Packets with a CRC error are posted to host memory only when *store-bad-packet* bit (RCTL.SBP) is set.
2. Hardware checks the data integrity when received packets are fetched from its internal packet buffer (see Section 7.6 for details). Packets with integrity errors are posted to host memory regardless of *store-bad-packet* setting (RCTL.SBP).

VLAN Tag Field (16)

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 RCTL.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 0x0000. The rule for VLAN tag is to use network ordering (also called big endian). It appears in the following manner in the descriptor:

Table 7-9. VLAN Tag Field Layout (for 802.1q Packet)

15 13	12	11 0
PRI	CFI	VLAN

7.1.5 Advanced Receive Descriptors

7.1.5.1 Advanced Receive Descriptors (Read Format)

Figure 7-10 shows the receive descriptor. This is the format that software writes to the descriptor queue and hardware reads from the descriptor queue in host memory. Hardware writes back the descriptor in a different format, shown in Table 7-10.

Table 7-10. Descriptor Read Format

	63 1	0
0	Packet Buffer Address [63:1]	A0/NSE
8	Header Buffer Address [63:1]	DD

Packet Buffer Address (64) - Physical address of the packet buffer. The lowest bit is either A0 (LSB of address) or NSE (No-Snoop Enable), depending on bit *RXCTL.RXdataWriteNSEn* of the relevant queue. See Section 8.13.1.

Header Buffer Address (64) - Physical address of the header buffer. The lowest bit is DD.

Note: The 82576 does not support null descriptors (a packet or header address is always equal to zero.

When software sets the *NSE* bit, the 82576 places the received packet associated with this descriptor in memory at the packet buffer address with *NSE* set in the PCIe attribute fields. NSE does not affect the data written to the header buffer address.

When a packet spans more than one descriptor, the header buffer address is not used for the second, third, etc. descriptors; only the packet buffer address is used in this case.

NSE is enabled for packet buffers that the software device driver knows have not been touched by the processor since the last time they were used, so the data cannot be in the processor cache and snoop is always a miss. Avoiding these snoop misses improves system performance. No-snoop is particularly useful when the DMA engine is moving the data from the packet buffer into application buffers, and the software device driver is using the information in the header buffer for its work with the packet.

Note: When No-Snoop Enable is used, relaxed ordering should also be enabled with CTRL_EXT.RO_DIS.

7.1.5.2 Advanced Receive Descriptors - Writeback Format

When the 82576 writes back the descriptors, it uses the descriptor format shown in Table 7-11.

Note: SRRCTL[n]. DESCTYPE must be set to a value other than 000b for the 82576 to write back the special descriptors.

Table 7-11. Descriptor Write-Back Format

	63 48	47 35	34 32	31	30 21	20	19 17	16 4	3 0
0	RSS Hash Value/Fragment Checksum and IP identification			SPH	HDR_LEN	RSV		Packet Type	RSS Type
8	VLAN Tag	PKT_LEN		Extended Error			Extended Status		

RSS Type (4)

Table 7-12. RSS Type

Packet Type	Description
0x0	No hash computation done for this packet.
0x1	HASH_TCP_IPV4
0x2	HASH_IPV4
0x3	HASH_TCP_IPV6
0x4	HASH_IPV6_EX
0x5	HASH_IPV6
0x6	HASH_TCP_IPV6_EX
0x7	HASH_UDP_IPV4
0x8	HASH_UDP_IPV6
0x9	HASH_UDP_IPV6_EX
0xA:0xF	Reserved

The 82576 must identify the packet type and then choose the appropriate RSS hash function to be used on the packet. The RSS type reports the packet type that was used for the RSS hash function.

Packet Type (13)

- VPKT (bit 12) - VLAN Packet indication

The 12 LSB bits of the packet type reports the packet type identified by the hardware as follows:

Table 7-13. LSB Bits

Bit Index	Bit 11 = 0b	Bit 11 = 1b (L2 packet)
0	IPV4 - IPv4 header present	EtherType - ETQF register index that matches the packet. Special types might be defined for 1588, 802.1X, or any other requested type.
1	IPV4E - IPv4 Header includes extensions	
2	IPV6 - IPv6 header present	
3	IPV6E- IPv6 Header includes extensions	Reserved - for future expansion of ETQF
4	TCP - TCP header present	
5	UDP - UDP header present	Reserved
6	SCTP - SCTP header present	Reserved
7	NFS - NFS header present	Reserved
8	IPSec ESP – IPSec encapsulation	Reserved
9	IPSec AH – IPSec encapsulation	Reserved
10	Reserved	Reserved

RSV(5) - Reserved.

HDR_LEN (10) - The length (bytes) of the header as parsed by the 82576. In split mode when HBO is set, the HDR_LEN can be greater then zero though nothing is written to the header buffer. In header replication mode (SPH is also set in this mode), this does not reflect the size of the data actually stored in the header buffer because the 82576 fills the buffer up to the size configured by SRRCTL[n].BSIZEHEADER, which might be larger than the header size reported here. This field is only valid in the first descriptor of a packet and should be ignored in all subsequent descriptors.

Packet types supported by the header split and header replication are listed later. Other packet types are posted sequentially in the host packet buffer. Each line in the following table has an enable bit in the PSRTYPE register. When one of the bits is set, the corresponding packet type is split. If the bit is not set, a packet matching the header layout is not split.

Header split and replication is described in Section 7.1.9 while the packet types for this functionality are enabled by the PSRTYPE[n] registers (Section 8.10.3).

Note: The header of a fragmented IPv6 packet is defined before the fragmented extension header.

SPH (1) - Split Header **-** When set, indicates that the *HDR_LEN* field reflects the length of the header found by hardware. If cleared, the *HDR_LEN* field should be ignored, unless in Split – always use header buffer mode, where PKT_LEN = 0, in which case, the HDR_LEN reflects the size of the packet, even if SPH is cleared.

RSS Hash / Fragment Checksum (32)

This field has multiplexed functionality according to the received packet type (reported on the *Packet Type* field in this descriptor) and device setting.

Fragment Checksum (16-Bit; 63:48)

The fragment checksum word contains the unadjusted one's complement checksum of the IP payload and is used to offload checksum verification for fragmented UDP packets as described in Section 7.1.10.2. This field is mutually exclusive with the RSS hash. It is enabled when the *RXCSUM.PCSD* bit is cleared and the *RXCSUM.IPPCSE* bit is set.

IP identification (16-Bit; 47:32)

The IP identification word identifies the IP packet to whom this fragment belongs and is used to offload checksum verification for fragmented UDP packets as described in Section 7.1.10.2. This field is mutually exclusive with the RSS hash. It is enabled when the *RXCSUM.PCSD* bit is cleared and the *RXCSUM.IPPCSE* bit is set.

RSS Hash Value (32)

The RSS hash value is required for RSS functionality as described in Section 7.1.1.7. This bit is mutually exclusive with the IP identification and the fragment checksum. It is enabled when the *RXCSUM.PCSD* bit is set.

Extended Status (20)

Status information indicates whether the descriptor has been used and whether the referenced buffer is the last one for the packet. Table 7-14 lists the extended status word in the last descriptor of a packet (*EOP* is set). Table 7-15 lists the extended status word in any descriptor but the last one of a packet (*EOP* is cleared).

Table 7-14. Receive Status (RDESC.STATUS) Layout of the Last Descriptor

19	18	17	16	15	14	13	12	11	10
Rsv	LB	SECP	TS	Reserved			Strip CRC	LLINT	UDPV

VEXT	Rsv	PIF	IPCS	L4I	UDPCS	VP	Rsv	EOP	DD
9	8	7	6	5	4	3	2	1	0

Table 7-15. Receive Status (RDESC.STATUS) Layout of Non-Last Descriptor

19	2	1	0
Reserved		EOP = 0b	DD

TS (16) - Time Stamped Packet (Time Sync). The Time Stamp bit is set when the device recognized a Time Sync packet. In such a case the hardware captures its arrival time and stores it in the "Time Stamp" register. For more details see Section 7.10.2.1.

Note: If TSYNCRXCTL.TYPE=100b, all the packets are time stamped; however, this bit is never set as the time stamp value is not locked.

Reserved (2, 8, 15:13, 19) - Reserved at zero.

PIF (7), IPCS(6), UDPCS(4), VP(3), EOP (1), DD (0) - These bits are described in the legacy descriptor format in Section 7.1.4.

L4I (5) - This bit indicates that an L4 integrity check was done on the packet, either TCP checksum, UDP checksum or SCTP CRC checksum. This bit is valid only for the last descriptor of the packet. An error in the integrity check is indicated by the *L4E* bit in the error field. The type of check done can be induced from the packet type bits 4, 5 and 6. If bit 4 is set, a TCP checksum was done. If bit 5 is set a UDP checksum was done, and if bit 6 is set, a CRC checksum was done.

VEXT (9)- First VLAN is found on a double VLAN packet. This bit is valid only when CTRL_EXT.EXTENDED_VLAN is set. For more details see Section 7.4.5.

UDPV (10) - This bit indicates that the incoming packet contains a valid (non-zero value) checksum field in an incoming fragmented UDP Ipv4 packet. This means that the *Fragment Checksum* field in the receive descriptor contains the UDP checksum as described in Section 7.1.10.2. When this field is cleared in the first fragment that contains the UDP header, 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.

LLINT (11)- This bit indicates that the packet caused an immediate interrupt via the low latency interrupt mechanism.

SECP (17) - The security processing bit indicates that hardware identified the security encapsulation and processed it as configured.

- IPsec processing: This bit is set only if a matched SA was found. Note that hardware does not process packets with the IPv4 option or IPv6 extension header and SECP is not set.

LB (18) - This bit provides a loopback status indication meaning that this packet is sent by a local virtual machine (VM-to-VM switch indication).

Extended Error (12)

Table 7-16 and the text that follows describes the possible errors reported by hardware.

Table 7-16. Receive Errors (RDESC.ERRORS) Layout

11	10	9	8 7	6 4	3	2 0
RXE	IPE	L4E	SECERR	Reserved	HBO	Reserved

RXE (bit 11) - RXE is described in the legacy descriptor format in Section 7.1.4.

IPE (bit 10) - The IPE error indication is described in the legacy descriptor format in Section 7.1.4.

L4E (bit 9) - L4 error indication - When set, indicates that hardware attempted to do an L4 integrity check as described in the *L4I* bit, but the check failed.

Security Error (bit 8:7)

IPSec Status (Valid Only if SA Match, Else Zero)

Indicates potential errors in the IPSec processing according to the following encoding:

Code	Error Type
00b	No error, either no SA match (SECP is cleared), or the incoming packet was successfully authenticated by hardware.[1]
01b	Invalid IPsec protocol, the *Protocol* field value included in the IP header (or in the IP next header for IPv6) does not match the *PROTO* field stored in the corresponding Rx SA entry.
10b	Packet length error, ESP packet is not 4-bytes aligned or the AH/ESP header is truncated (for example, a 28-byte IPv4 packet with IPv4 header and ESP header that contains only SPI and SN) or AH Length field content in AH header is not valid (i.e. not equal to 0x07 for IPv4 or to 0x08 for IPv6).
11b	Authentication failed. For example, the computed ICV field does not match the ICV field included in the packet.

1. For incoming IPv4 packets where the Protocol field is AH/ESP, and for which any IPv4 option is present; or for incoming IPv6 packets where there is an AH/ESP extension header together with any other extension header (even another AH/ESP extension header), no IPsec or Layer4 offload is performed by hardware and the packet is passed to software with the SECP bit cleared (such as no SA match) - without performing any SA lookup.

Reserved (bit 6:4)

HBO (bit 3) - Header Buffer Overflow

Note: This bit is relevant only if *SPH* is set.

1. In both header replication modes, *HBO* is set if the header size (as calculated by hardware) is bigger than the allocated buffer size (*SRRCTL.BSIZEHEADER*) but the replication still takes place up to the header buffer size. Hardware sets this bit in order to indicate to software that it needs to allocate bigger buffers for the headers.
2. In header split mode, when *SRRCTL[n] BSIZEHEADER* is smaller than *HDR_LEN*, then *HBO* is set to 1b, In this case, the header is not split. Instead, the header resides within the host packet buffer. The *HDR_LEN* field is still valid and equal to the calculated size of the header. However, the header is not copied into the header buffer.
3. In header split mode, always use header buffer mode, when *SRRCTL[n] BSIZEHEADER* is smaller than *HDR_LEN*, then *HBO* is set to 1b. In this case, the header buffer is used as part of the data buffers and contains the first BSIZEHEADER bytes of the packet. The *HDR_LEN* field is still valid and equal to the calculated size of the header.

Note: Most error information appears only when the *store–bad–packet* bit (*RCTL.SBP*) is set and a bad packet is received.

Reserved (bits 2:0) - Reserved

PKT_LEN (16) – Number of bytes existing in the host packet buffer

The length covers the data written to a receive buffer including CRC bytes (if any). Software must read multiple descriptors to determine the complete length for packets that span multiple receive buffers. If SRRCTL.DESC_TYPE = 4 (advanced descriptor header replication large packet only) and the total packet length is smaller than the size of the header buffer (no replication is done), this field continues to reflect the size of the packet, although no data is written to the packet buffer. Otherwise, if the buffer is not split because the header is bigger than the allocated header buffer, this field reflects the size of the data written to the first packet buffer (header and data).

VLAN Tag (16)

These bits are described in the legacy descriptor format in Section 7.1.4.

7.1.6 Receive Descriptor Fetching

The fetching algorithm attempts to make the best use of PCIe bandwidth by fetching a cache-line (or more) descriptor 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 (host writes to the tail pointer). When the on-chip buffer is nearly empty (RXDCTL.PTHRESH), a prefetch is performed each time enough valid descriptors (RXDCTL.HTHRESH) are available in host memory.

When the number of descriptors in host memory is greater than the available on-chip descriptor storage, the 82576 might elect to perform a fetch that is not a multiple of cache-line size. Hardware performs this non-aligned fetch if doing so results in the next descriptor fetch being aligned on a cache-line boundary. This enables the descriptor fetch mechanism to be most efficient in the cases where it has fallen behind software.

All fetch decisions are based on the number of descriptors available and do not take into account any split of the transaction due to bus access limitations.

Note: The 82576 NEVER fetches descriptors beyond the descriptor tail pointer.

7.1.7 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. A receive descriptor packing mechanism minimizes the occurrence of partial line write-backs.

To maximize memory efficiency, receive descriptors are packed together and written as a cache-line whenever possible. Descriptors write-backs accumulate and are opportunistically written out in cache line-oriented chunks, under the following scenarios:

- RXDCTL.WTHRESH descriptors have been used (the specified maximum threshold of unwritten used descriptors has been reached).
- The receive timer expires (EITR) - in this case all descriptors are flushed ignoring any cache-line boundaries.
- Explicit software flush (RXDCTLn.SWFLS).
- Dynamic packets - if at least one of the descriptors that are waiting for write-back are classified as packets requiring immediate notification the entire queue is flushed out.

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 the receive timer (EITR) expires, all used descriptors are forced to be written back prior to initiating the interrupt, for consistency. Software might explicitly flush accumulated descriptors by writing the RXDCTLn register with the *SWFLS* bit set.

When the 82576 does a partial cache-line write-back, it attempts to recover to cache-line alignment on the next write-back.

For applications where the latency of received packets is more important that the bus efficiency and the CPU utilization, an EITR value of zero may be used. In this case, each receive descriptor will be written to the host immediately. If RXDCTL.WTHRESH equals zero, then each descriptor will be written back separately, otherwise, write back of descriptors may be coalesced if descriptor accumulates in the internal descriptor ring due to bandwidth constrains.

All write-back decisions are based on the number of descriptors available and do not take into account any split of the transaction due to bus access limitations.

7.1.8 Receive Descriptor Ring Structure

Figure 7-9 shows the structure of each of the 16 receive descriptor rings. Hardware maintains 16 circular queues of descriptors and writes back used descriptors just prior to advancing the head pointer(s). Head and tail pointers wrap back to base when size descriptors have been processed.

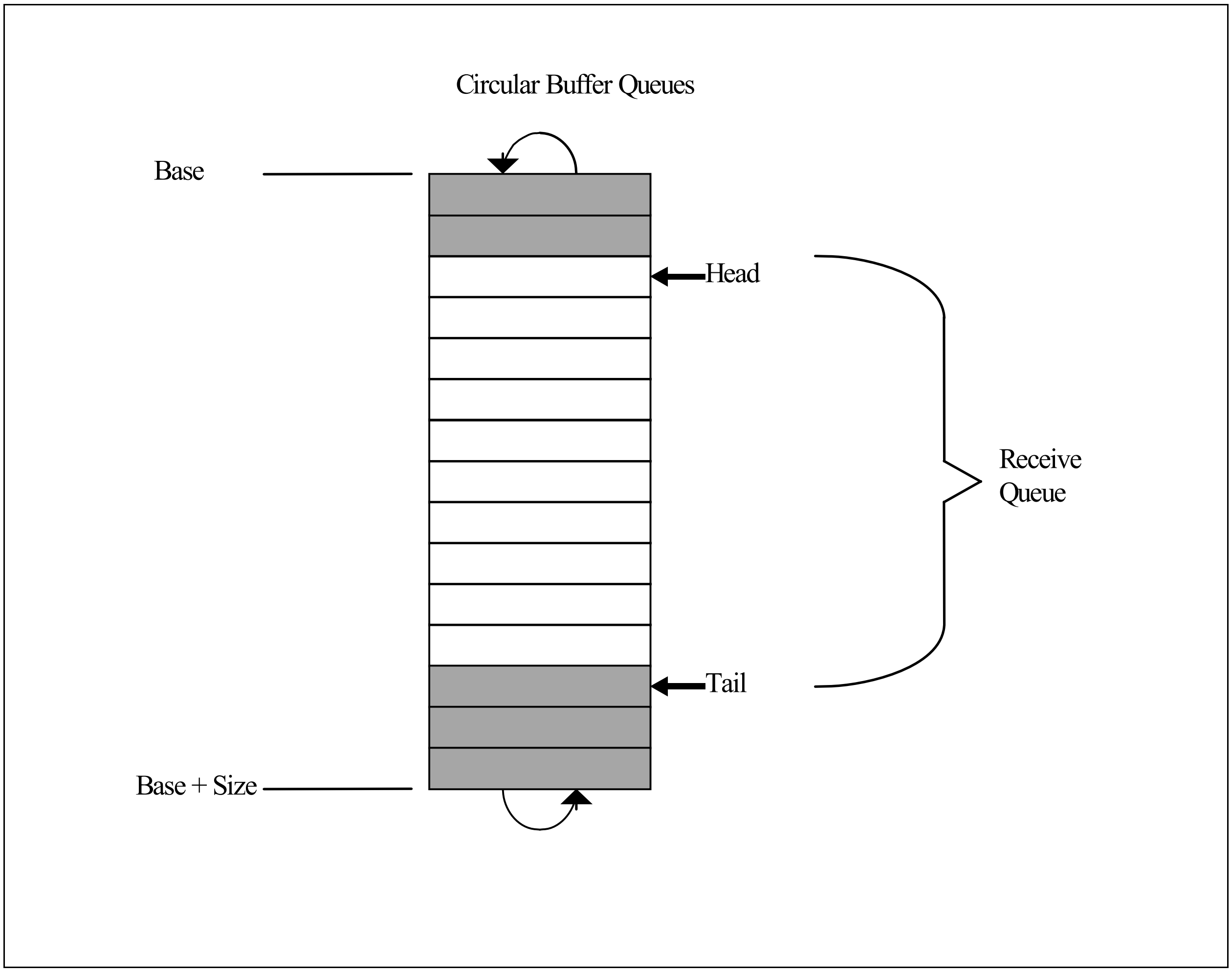


Figure 7-9. Receive Descriptor Ring Structure

Software inserts receive descriptors by advancing the tail pointer(s) to refer to the address of the entry just beyond the last valid descriptor. This is accomplished by writing the descriptor tail register(s) with the offset of the entry beyond the last valid descriptor. The hardware adjusts its internal tail pointer(s) accordingly. As packets arrive, they are stored in memory and the head pointer(s) is incremented by hardware. When the head pointer(s) is equal to the tail pointer(s), the queue(s) is empty. Hardware stops storing packets in system memory until software advances the tail pointer(s), making more receive buffers available.

The receive descriptor head and tail pointers reference to 16-byte blocks of memory. Shaded boxes in Figure 7-9 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 the descriptors in memory. Any descriptor with a non-zero DD value has been processed by the hardware and is ready to be handled by the software.

Note: The head pointer points to the next descriptor that is written back. After the descriptor write-back operation completes, this pointer is incremented by the number of descriptors written back. Hardware owns all descriptors between [head.. tail]. Any descriptor not in this range is owned by software.

The receive descriptor rings are described by the following registers:

- Receive Descriptor Base Address (RDBA15 to RDBA0) registers:
  This register indicates 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. Note that hardware ignores the lower 4 bits.
- Receive Descriptor Length (RDLEN15 to RDLEN0) registers:
  This register determines the number of bytes allocated to the circular buffer. This value must be a multiple of 128 (the maximum cache-line size). Since each descriptor is 16 bytes in length, the total number of receive descriptors is always a multiple of eight.
- Receive Descriptor Head (RDH15 to RDH0) registers:
  This register holds a value that is an offset from the base and indicates the in-progress descriptor. There can be up to 64 KB, 8 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 (RDT15 to RDT0) registers:
  This register holds a value that is an offset from the base and identifies the location beyond the last descriptor hardware can process. This is the location where software writes the first new descriptor.

If software statically allocates buffers, uses legacy receive descriptors, and uses memory read to check for completed descriptors, it has to zero the status byte in the descriptor before bumping the tail pointer to make it ready for reuse by hardware. Zeroing the status byte is not a hardware requirement but is necessary for performing an in-memory scan.

All the registers controlling the descriptor rings behavior should be set before receive is enabled, apart from the tail registers that are used during the regular flow of data.

7.1.8.1 Low Receive Descriptors Threshold

As described above, the size of the receive queues is measured by the number of receive descriptor. During run time the software processes completed descriptors and then increments the Receive Descriptor Tail registers (RDT). At the same time, the hardware may post new packets received from the LAN incrementing the Receive Descriptor Head registers (RDH) for each used descriptor.

The number of usable (free) descriptors for the hardware is the distance between Tail and Head registers. When the Tail reaches the Head, there are no free descriptors and further packets may be either dropped or block the receive FIFO. In order to avoid it, the 82576 may generate a low latency interrupt (associated to the relevant Rx queue) once there are less equal free descriptors than a threshold. The threshold is defined in 16 descriptors granularity per queue in the SRRCTL[n].RDMTS field.

7.1.9 Header Splitting and Replication

7.1.9.1 Purpose

This feature consists of splitting or replicating packet's header to a different memory space. This helps the host to fetch headers only for processing: headers are replicated through a regular snoop transaction in order to be processed by the host CPU. It is recommended to perform this transaction with the DCA feature enabled (see section 8.3) or in conjunction with a software-prefetch.

The packet (header and payload) is stored in memory through a (optionally) non-snoop transaction. Later, a data movement engine transaction moves the payload from the software device driver buffer to application memory or it is moved using a normal memory copy operation.

The 82576 supports header splitting in several modes:

- Legacy mode: legacy descriptors are used; headers and payloads are not split.
- Advanced mode, no split: advanced descriptors are in use; header and payload are not split.
- Advanced mode, split: advanced descriptors are in use; header and payload are split to different buffers. If the packet cannot be split, only the packet buffer is used.
- Advanced mode, replication: advanced descriptors are in use; header is replicated in a separate buffer and also in a payload buffer.
- Advanced mode, replication, conditioned by packet size: advanced descriptors are in use; replication is performed only if the packet is larger than the header buffer size.
- Advanced mode, split, always use header buffer: advanced descriptors are in use; header and payload are split to different buffers. If no split is done, the first part of the packet is stored in the header buffer.

7.1.9.2 Description

In Figure 7-10 and Figure 7-11, the header splitting and header replication modes are shown.

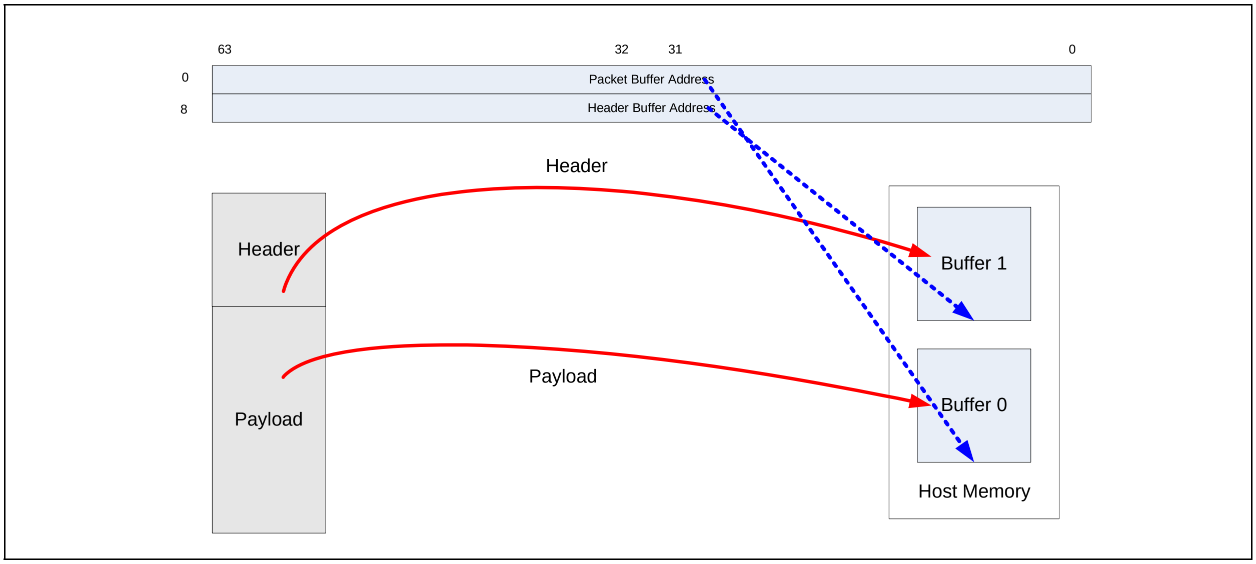


Figure 7-10. Header Splitting

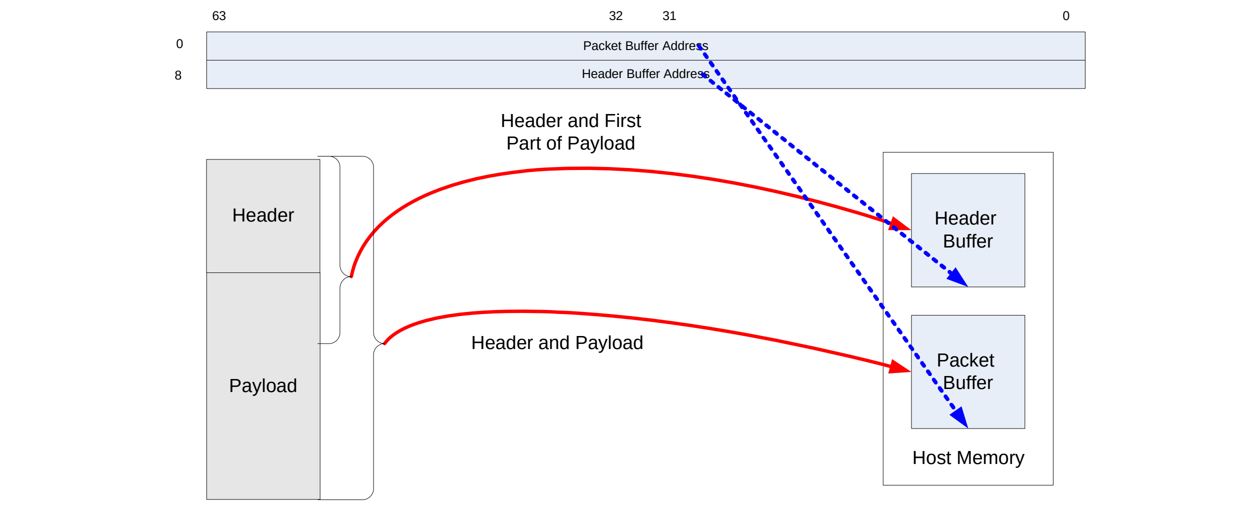


Figure 7-11. Header Replication

The physical address of each buffer is written in the *Buffer Addresses* fields. The sizes of these buffers are statically defined by BSIZEPACKET in the SRRCTL[n] registers.

The packet buffer address includes the address of the buffer assigned to the replicated packet, including header and data payload portions of the received packet. In the case of a split header, only the payload is included.

The header buffer address 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 82576 uses the packet replication or splitting feature when the SRRCTL[n].DESCTYPE is larger that one. The software device driver must also program the buffer sizes in the SRRCTL[n] registers.

When header split is selected, the packet is split only on selected types of packets. A bit exists for each option in PSRTYPE[n] registers so several options can be used in conjunction with them. If one or more bits are set, the splitting is performed for the corresponding packet type.

The following table lists the behavior of the 82576 in the different modes:

Table 7-17. Intel® 82576 GbE Controller Behavior

DESCTYPE	Condition	SPH	HBO	PKT_LEN	HDR_LEN	Header and Payload DMA
Split	1. Header can't be decoded	0b	0b	Min(Packet length, Buffer size)	N/A	Header + Payload → Packet buffer
	2. Header <= BSIZEHEADER	1b	0b	Min(Payload length, Buffer size)[1]	Header size	Header → Header buffer Payload → Packet buffer
	3. Header > BSIZEHEADER	1b	1b	Min(Packet length, Buffer size)	Header size[2]	Header + Payload → Packet buffer

Table 7-17. Intel® 82576 GbE Controller Behavior

Split – always use header buffer	1. Packet length <= BSIZEHEADER	0b	0b	Zero	Packet length	Header + Payload → Header buffer
	2. Header can't be Decoded and Packet length > BSIZEHEADER	0b	0b	Min(Packet length – BSIZEHEADER, Data Buffer size)	BSIZEHEADER	Header + Payload → Header + Packet buffers[3]
	3. Header <= BSIZEHEADER and Packet length >= BSIZEHEADER	1b	0b	Min(Payload length, Data Buffer size)	Header Size	Header → Header buffer Payload → Packet buffer
	4. Header > BSIZEHEADER	1b	1b	Min(Packet length – BSIZEHEADER, Data Buffer size)	Header Size[2]	Header + Payload → Header + Packet buffer[3]
Replicate Large Packet only	1. Header + Payload <= BSIZEHEADER	0b/1b[4]	0b	Packet length	Header size, N/A[4]	Header + Payload → Header buffer
	2. Header + Payload > BSIZEHEADER	0b/1b[4]	0b/1b[5]	Min(Packet length, Buffer size)	Header size, N/A[4]	(Header + Payload)(partial[6]) → Header buffer Header + Payload → Packet buffer

1. In a header only packet (such as TCP ACK packet), the PKT_LEN is zero.
2. The HDR_LEN doesn't reflect the actual data size stored in the Header buffer. It reflects the header size determined by the parser.
3. If the packet spans more than one descriptor, only the header buffer of the first descriptor is used. The header buffer is used for the first part of the packet until it is filled up, and then the first packet buffer is used for the continuation of the packet.

Software Notes:

- If SRRCTL#.NSE is set, all buffers' addresses in a packet descriptor must be word aligned.
- Packet header can't span across buffers, therefore, the size of the header buffer must be larger than any expected header size. Otherwise, only the part of the header fitting the header buffer is replicated. In the case of header split mode (SRRCTL.DESCTYPE = 010b), a packet with a header larger than the header buffer is not split.

7.1.10 Receive Packet Checksum Off Loading

The 82576 supports the off loading of three receive checksum calculations: the packet checksum, the IPv4 header checksum, and the TCP/UDP checksum.

The packet checksum is the one's complement over the receive packet, starting from the byte indicated by RXCSUM.PCSS (zero corresponds to the first byte of the packet), after stripping. For packets with a VLAN header, the packet checksum includes the header if VLAN striping is not enabled by the CTRL.VME. If a 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 and CTRL.VME is set 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 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 82576. If *RXCSUM.IPOFL* is set to 1b, the 82576 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.TUOFL* is set to 1b, the 82576 calculates the TCP or UDP checksum and indicates a pass/fail condition to software via the TCP/UDP *Checksum Error* bit (*RDESC.L4E*). These error bits are valid when the respective status bits indicate the checksum was calculated for the packet (*RDESC.IPCS* and *RDESC.L4CS*, respectively). Similarly, if *RFCTL.Ipv6_DIS* and *RFCTL.IP6Xsum_DIS* are cleared to 0b and *RXCSUM.TUOFL* is set to 1b, the 82576 calculates the TCP or UDP checksum for IPv6 packets. It then indicates a pass/fail condition in the TCP/UDP *Checksum Error* bit (*RDESC.L4E*).

If neither *RXCSUM.IPOFL* nor *RXCSUM.TUOFL* are set, the *Checksum Error* bits (IPE and L4E) are 0b for all packets.

Supported frame types:

- Ethernet II
- Ethernet SNAP

7.1.10.1 Filters details

Table 7-18. Supported Receive Checksum Capabilities

Packet Type	Hardware IP Checksum Calculation	Hardware 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 (with len zero) • Routing (with len > zero) • Fragment • Home option	 No (n/a) No (n/a) No (n/a) No (n/a) No (n/a) No (n/a)	 Yes Yes Yes No No No
IPv4 tunnels: • IPv4 packet in an IPv4 tunnel. • IPv6 packet in an IPv4 tunnel.	 No Yes (IPv4)	 No Yes[1]
IPv6 tunnels: • IPv4 packet in an IPv6 tunnel. • IPv6 packet in an IPv6 tunnel.	 No No	 No No
Packet is an Ppv4 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 number other than TCP or UDP.	Yes	No

[1] The IPv6 header portion can include supported extension headers as described in the IPv6 filter section.

The previous table lists general details about what packets are processed. In more detail, the packets are passed through a series of filters to determine if a receive checksum is calculated:

7.1.10.1.1 MAC Address Filter

This filter checks the MAC destination address to be sure it is valid (such as 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.

7.1.10.1.2 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.

7.1.10.1.3 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 five (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.

7.1.10.1.4 IPv6 Filter

This filter checks for valid IPpv6 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).

7.1.10.1.5 IPv6 Extension Headers

IPv4 and TCP provide header lengths, which enable hardware to easily navigate through these headers on packet reception for calculating checksum and CRCs, etc. For receiving IPv6 packets; however, there is no IP header length to help hardware find the packet's ULP (such as TCP or UDP) header. One or more IPv6 extension headers might exist in a packet between the basic IPv6 header and the ULP header. The hardware must skip over these extension headers to calculate the TCP or UDP checksum for received packets.

The IPv6 header length without extensions is 40 bytes. The IPv6 field *Next Header Type* indicates what type of header follows the IPv6 header at offset 40. It might be an upper layer protocol header such as TCP or UDP (*Next Header Type* of 6 or 17, respectively), or it might indicate that an extension header follows. The final extension header indicates with its *Next Header Type* field the type of ULP header for the packet.

IPv6 extension headers have a specified order. However, destinations must be able to process these headers in any order. Also, IPv6 (or IPv4) might be tunneled using IPv6, and thus another IPv6 (or IPv4) header and potentially its extension headers might be found after the extension headers.

The IPv4 *Next Header Type* is at byte offset nine. In IPv6, the first *Next Header Type* is at byte offset six.

All IPv6 extension headers have the *Next Header Type* in their first eight bits. Most have the length in the second eight bits (Offset Byte[1]) as shown:

Table 7-19. Typical IPv6 Extended Header Format (Traditional Representation)

	1	2 3
0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Next Header Type	Length	

The following table lists the encoding of the *Next Header Type* field and information on determining each header type's length. The IPv6 extension headers are not otherwise processed by the 82576 so their details are not covered here.

Table 7-20. Header Type Encoding and Lengths

Header	Next Header Type	Header Length (Units are Bytes Unless Otherwise Specified)
IPv6	6	Always 40 bytes
IPv4	4	Offset Bits[7:4] Unit = 4 bytes
TCP	6	Offset Byte[12].Bits[7:4] Unit = 4 bytes
UDP	17	Always 8 bytes
Hop by Hop Options	0 (Note 1)	8+Offset Byte[1]
Destination Options	60	8+Offset Byte[1]
Routing	43	8+Offset Byte[1]
Fragment	44	Always 8 bytes
Authentication	51	8+4*(Offset Byte[1])
Encapsulating Security Payload	50	Note 3
No Next Header	59	Note 2

Notes:

1. Hop-by-hop options header is only found in the first *Next Header Type* of an IPv6 header.
2. When a *No Next Header* type is encountered, the rest of the packet should not be processed.
3. Encapsulated security payload - Intel® 82576 GbE Controller cannot offload packets with this header type.

Note that the 82576 hardware acceleration does not support all IPv6 extension header types (refer to Table 7-20).

Also, the *RFCTL.Ipv6_DIS* bit must be cleared for this filter to pass.

7.1.10.1.6 UDP/TCP Filter

This filter checks for a valid UDP or TCP header. The prototype next header values are 0x11 and 0x06, respectively. The *RXCSUM.TUOFL* bit must be set for this filter to pass.

7.1.10.2 Receive UDP Fragmentation Checksum

The 82576 might provide receive fragmented UDP checksum offload. The 82576 should be configured in the following manner to enable this mode:

The *RXCSUM.PCSD* bit should be cleared. The *Packet Checksum* and *IP Identification* fields are mutually exclusive with the RSS hash. When the *PCSD* bit is cleared, *Packet Checksum* and *IP Identification* are active instead of RSS hash.

The *RXCSUM.IPPCSE* bit should be set. This field enables the IP payload checksum enable that is designed for the fragmented UDP checksum.

The *RXCSUM.PCSS* field must be zero. The packet checksum start should be zero to enable auto-start of the checksum calculation. The following table lists the exact description of the checksum calculation.

The following table also lists the outcome descriptor fields for the following incoming packets types:

Table 7-21. Descriptor Fields

Incoming Packet Type	Fragment Checksum	UDPV	UDPCS / L4CS
Non IP Packet	0b	0b	0b / 0b
Ipv6 Packet	0b	0b	Depends on transport header.
Non fragmented Ipv4 packet	0b	0b	Depends on transport header.
Fragmented Ipv4, when not first fragment	The unadjusted one's complement checksum of the IP payload.	0b	1b / 0b
Fragmented Ipv4, for the first fragment	Same as above	1 if the UDP header checksum is valid (not zero)	1b / 0b

Note: When the software device driver computes the 16-bit ones complement, the sum on the incoming packets of the UDP fragments, it should expect a value of 0xFFFF. Refer to Section 7.1.10 for supported packet formats.

7.1.11 SCTP Offload

If a receive packet is identified as SCTP, the 82576 checks the CRC32 checksum of this packet and identifies this packet as SCTP. Software is notified of the CRC check via the *CRCV* bit in the *Extended Status* field of the Rx descriptor. The detection of an SCTP packet is indicated via the *SCTP* bit in the packet *Type* field of the Rx descriptor. The checker assumes the following SCTP packet format:

Table 7-22. SCTP Header

	1	2	3
0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3	4 5 6 7 8 9 0 1
Source Port		Destination Port	
Verification Tag			
Checksum			
Chunks 1..n			

7.2 Transmit Functionality

7.2.1 Packet Transmission

Output packets are made up of pointer-length pairs constituting a descriptor chain (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 the L2 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 under-runs.

Another transmit feature of the 82576 is TCP/UDP segmentation. The hardware has the capability to perform packet segmentation on large data buffers offloaded from the Network Operating System (NOS). This feature is discussed in detail in Section 7.2.4.

In addition, the 82576 supports SCTP offloading for transmit requests. See section Section 7.2.5.3 for details about SCTP.

7.2.1.1 Transmit Data Storage

Data is 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 (9728 bytes). A packet typically consists of two (or more) buffers, one (or more) for the header and one for the actual data. Each buffer is referenced by a different descriptor. Some software implementations copy the header(s) and packet data into one buffer and use only one descriptor per transmitted packet.

7.2.1.2 On-Chip Tx Buffers

The 82576 contains a 40 KB packet buffer that can be used to store packets until they are forwarded to the network or locally to another Virtual Machine (VM).

7.2.1.3 On-Chip descriptor Buffers

The 82576 contains a 32 descriptor cache for each transmit queue used to reduce the latency of packet processing and to optimize the usage of the PCIe bandwidth by fetching and writing back descriptors in bursts. The fetch and writeback algorithm are described in Section 7.2.2.5 and Section 7.2.2.6.

7.2.1.4 Transmit Contexts

The 82576 provides hardware checksum offload and TCP/UDP segmentation facilities. 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. Part of the parameters used by these features is handled though contexts.

A context refers to a set of device registers loaded or accessed as a group to provide a particular function. The 82576 supports 32 context register sets on-chip (two per queue). The transmit queues can contain transmit data descriptors (much like the receive queue) as well as transmit context descriptors.

The contexts are queue specific and one context cannot be reused from one queue to another. This differs from the method used in previous devices that supported a pool of contexts to be shared between queues.

A transmit context descriptor differs from a data descriptor as it does not point to packet data. Instead, this descriptor provides the ability to write to the on-chip contexts that support the transmit checksum offloading and the segmentation features of the 82576.

The 82576 supports one type of transmit context. This on-chip context is written with a transmit context descriptor DTYP=2 and is always used for transmit data descriptor DTYP=3.

The *IDX* field contains an index to one of the two queue contexts. Software must track what context is stored in each *IDX* location.

Each advanced data descriptor that uses any of the advanced offloading features must refer to a context.

Contexts can be initialized with a transmit context descriptor and then used for a series of related transmit data descriptors. The context, for example, defines the checksum and offload capabilities for a given type of TCP/IP flow. All packets of this type can be sent using this context.

Software is responsible for ensuring that a context is only overwritten when it is no longer needed. Hardware does not include any logic to manage the on-chip contexts; it is completely up to software to populate and then use the on-chip context table.

Each context defines information about the packet sent including the total size of the MAC header (TDESC.MACHDR), the amount of payload data that should be included in each packet (TDESC.MSS), TCP header length (TDESC.TCPHDR), IP header length (TDESC.IPHDR), and information about what type of protocol (TCP, IP, etc.) is used. Other than TCP, IP (TDESC.TUCMD), most information is specific to the segmentation capability.

Because there are dedicated on-chip resources for contexts, they remain constant until they are modified by another context descriptor. This means that a context can be used for multiple packets (or multiple segmentation blocks) unless a new context is loaded prior to each new packet. Depending on the environment, it might be 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. This new context could use a different index or the same index.

This same logic can also be applied to the TCP/UDP segmentation scenario, 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 context could be used for multiple TCP messages that require hardware segmentation.

7.2.2 Transmit Descriptors

The 82576 supports legacy descriptors and the 82576 advanced descriptors.

Legacy descriptors are intended to support legacy drivers to enable fast platform power up and to facilitate debug.

Note: These descriptors should not be used when advanced features such as virtualization. If legacy descriptors are used when *CTRL_EXT.RT* or *DTXSWC.Loopback enable* or *STATUS.VFE* or one of the *DTXSWC.MACAS* bits or one of the *DTXSWC.VLANAS* bits are set, the packets are ignored and not sent.

The Legacy descriptors are recognized as such based on the *DEXT* bit as discussed later in this section.

In addition, the 82576 supports two types of advanced transmit descriptors:

1. Advanced Transmit Context Descriptor, DTYP = 0010b.
2. Advanced Transmit Data Descriptor, DTYP = 0011b.

Note: DTYP values 0000b and 0001b are reserved.

The transmit data descriptor (both legacy and advanced) points to a block of packet data to be transmitted. The advanced transmit context descriptor does not point to packet data. It contains control/context information that is loaded into on-chip registers that affect the processing of packets for transmission. The following sections describe the descriptor formats.

7.2.2.1 Legacy Transmit Descriptor Format

Legacy descriptors are identified by having bit 29 of the descriptor (*TDESC.DEXT*) set to 0b. In this case, the descriptor format is defined as shown in Table 7-23. Note that the address and length must be supplied by software. Also note that bits in the command byte are optional, as are the CSO, and CSS fields.

Table 7-23. Transmit Descriptor (TDESC) Fetch Layout - Legacy Mode

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

Table 7-24. Transmit Descriptor (TDESC) Write-Back Layout - Legacy Mode

	63 48	47 40	39 36	35 32	31 24	23 16	15 0
0	Reserved				Reserved		
8	VLAN	CSS	Reserved	STA	CMD	CSO	Length

Note: For frames that spans multiple descriptors, the VLAN, CSS, CSO, CMD.VLE, CMD.IC, and CMD.IFCS are valid only in the first descriptors and are ignored in the subsequent ones.

7.2.2.1.1 Length

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 9728 bytes.

Note: The maximum allowable packet size for transmits changes based on the value written to the Packet Buffer Allocation (PBA) register.

Descriptor length(s) might 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 9728 bytes.

Note: Descriptors with zero length (null descriptors) transfer no data. Null descriptors can only appear between packets and must have their *EOP* bits set.

If the *TCTL.PSP* bit is set, the total length of the packet transmitted, not including FCS should be at least 17 bytes.

7.2.2.1.2 Checksum Offset and Start - CSO and CSS

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 and must be in the range of data provided to the 82576 in the descriptor. This means for short packets that are not padded by software, CSS and CSO must be in the range of the unpadded data length, not the eventual padded length (64 bytes). CSO must be larger than CSS, CSS must be equal or greater than 14 bytes, and CSO must be smaller than the packet length minus four bytes. Checksum calculation is not done if CSO or CSS are out of range. This occurs if (CSS > length) OR (CSO > length - 1).

In the case of an 802.1Q header, the offset values depend on the VLAN insertion enable (*VLE*) bit. If they are not set (VLAN tagging included in the packet buffers), the offset values should include the VLAN tagging. If these bits are set (VLAN tagging is taken from the packet descriptor), the offset values should exclude the VLAN tagging.

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 to be inserted. 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.

UDP checksum calculation is not supported by the legacy descriptor as when using legacy descriptors. The 82576 is not aware of the L4 type of the packet and thus, does not support the translation of a checksum result of 0x0000 to 0xFFFF needed to differentiate between an UDP packet with a checksum of zero and an UDP packet without checksum.

Because the *CSO* field is eight bits wide, it puts a limit on the location of the checksum to 255 bytes from the beginning of the packet.

Hardware adds the checksum to the field at the offset indicated by the *CSO* field. Checksum calculations are for the entire packet starting at the byte indicated by the *CSS* field. A value of zero corresponds to the first byte in the packet.

CSS must be set in the first descriptor for a packet.

Table 7-25. Transmit Command (TDESC.CMD) Layout

7	6	5	4	3	2	1	0
RSV	VLE	DEXT	Rsv	RS	IC	IFCS	EOP

7.2.2.1.3 Command Byte - CMD

The CMD byte stores the applicable command and has the fields shown in Figure 7-25.

- RSV (bit 7) - Reserved
- VLE (bit 6) - VLAN Packet Enable
- DEXT (bit 5) - Descriptor Extension (0 for legacy mode)
- Reserved (bit 4) - Reserved
- RS (bit 3) - Report Status
- IC (bit 2) - Insert Checksum
- IFCS (bit 1) - Insert FCS
- EOP (bit 0) - End of Packet

VLE: Indicates that the packet is a VLAN packet. For example, hardware should add the VLAN Ethertype and an 802.1q VLAN tag to the packet.

RS: Signals the hardware to report the status information. This is used by software that does in-memory checks of the transmit descriptors to determine which ones are done. For example, if software queues up 10 packets to transmit, it can set the *RS* bit in the last descriptor of the last packet. If software maintains a list of descriptors with the *RS* bit set, it can look at them to determine if all packets up to (and including) the one with the *RS* bit set have been buffered in the output FIFO. Looking at the status byte and checking the *Descriptor Done* (*DD*) bit do this. If *DD* is set, the descriptor has been processed. Refer to Figure 7-27 for the layout of the status field.

IC: If set, requests hardware to add the checksum of the data from *CSS* to the end of the packet at the offset indicated by the *CSO* field.

IFCS: When set, hardware appends the MAC FCS at the end of the packet. When cleared, software should calculate the FCS for proper CRC check. There are several cases in which software must set IFCS:

- Transmitting a 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 or by the VMVIR registers.

EOP, when set, indicates the last descriptor making up the packet. Note that one or many descriptors can be used to form a packet.

Note: As opposed to 82571EB: VLE, IFCS, CSO, and IC must be set correctly in the first descriptor of each packet. In previous silicon generations, some of these bits were required to be set in the last descriptor of a packet.

Table 7-26. VLAN Tag Insertion Decision Table

VLE	Action
0b	Send generic Ethernet packet.
1b	Send 802.1Q packet; the Ethernet *Type* field comes from the VET register and the VLAN data comes from the *VLAN* field of the TX descriptor;

7.2.2.1.4 Status – STA

One bit provides transmit status, when *RS* is set in the command: *DD* indicates that the descriptor is done and is written back after the descriptor has been processed.

Note: When head write-back is enabled, the write-back of the *DD* bit to the descriptor is not executed.

Table 7-27. Transmit Status (TDESC.STA) Layout

3	2	1	0
Reserved			DD

7.2.2.1.5 DD (Bit 0) - Descriptor Done Status

7.2.2.1.6 VLAN

The *VLAN* field is used to provide the 802.1q/802.1ac tagging information. The *VLAN* field is qualified only on the first descriptor of each packet when the *VLE* bit is set. The rule for VLAN tag is to use network ordering (also called big endian). It appears in the following manner in the descriptor:

Table 7-28. VLAN Field (TDESC.VLAN) Layout

15 13	12	11 0
PRI	CFI	VLAN ID

- VLAN ID - the 12-bit tag indicating the VLAN group of the packet.
- Canonical Form Indication (CFI) - Set to zero for Ethernet packets.
- PRI - indicates the priority of the packet.

Note: The VLAN tag should be sent in network order.

7.2.2.2 Advanced Transmit Context Descriptor

Table 7-29. Transmit Context Descriptor (TDESC) Layout - (Type = 0010b)

	63 40	39 32	31 16	15 9	8 0
0	Reserved	IPsec SA Index	VLAN	MACLEN	IPLEN

	63 48	47 40	39	38 36	35 30	29	28 24	23 20	19 9	8 0
8	MSS	L4LEN	RSV	IDX	Reserved	DEXT	RSV	DTYP	TUCMD	IPSec ESP_LEN

7.2.2.2.1 IPLEN (9)

IP header length. If an offload is requested, IPLEN must be greater than or equal to 20 and less than or equal to 511. For IPsec flows, it includes the length of the IPsec header.

7.2.2.2.2 MACLEN (7)

This field indicates the length of the MAC header. When an offload is requested (one of TSE or IXSM or TXSM is set), MACHDR must be larger than or equal to 14 and less than or equal to 127. This field should include only the part of the L2 header supplied by the software device driver and not the parts added by hardware. The following table lists the value of MACLEN in the different cases.

Table 7-30. MACLEN Values

SNAP	Regular VLAN	Extended VLAN	MACLEN
No	By hardware or no	No	14
No	By hardware or no	Yes	18
No	By software	No	18
No	By software	Yes	22
Yes	By hardware or no	No	22
Yes	By hardware or no	Yes	26
Yes	By software	No	26
Yes	By software	Yes	30

VLAN (16) - 802.1Q VLAN tag to be inserted in the packet during transmission. This VLAN tag is inserted and needed only when a packet using this context has its *DCMD.VLE* bit set. This field should include the entire 16-bit *VLAN* field including the *CFI* and *Priority* fields as shown in Figure 7-28.

Note: The VLAN tag should be sent in network order.

7.2.2.2.3 IPsec SA IDX (8)

IPsec SA Index. If an IPsec offload is requested for the packet (IPSEC bit is set in the advanced Tx data descriptor), indicates the index in the SA table where the IPsec key and SALT are stored for that flow.

7.2.2.2.4 Reserved (24)

7.2.2.2.5 IPS_ESP_LEN (9)

Size of the ESP trailer and ESP ICV appended by software. Meaningful only if the IPSEC_TYPE bit is set in the TUCMD field and to single send packets for which the IPSEC bit is set in their advanced Tx data descriptor.

7.2.2.2.6 TUCMD (11)

- RSV (bit 10-6) - Reserved
- Encryption (bit5) - ESP encryption offload is required. Meaningful only to packets for which the IPSEC bit is set in their advanced Tx data descriptor.
- IPSEC_TYPE (bit 4) - Set for ESP. Cleared for AH. Meaningful only to packets for which the IPSEC bit is set in their advanced Tx data descriptor.
- L4T (bit 3:2) - L4 Packet TYPE (00b: UDP; 01b: TCP; 10b: SCTP; 11b: RSV)
- IPV4 (bit 1) - IP Packet Type: When 1b, Ipv4; when 0b, Ipv6
- SNAP (bit 0) - SNAP indication

7.2.2.2.7 DTYP (4)

Always 0010b for this type of descriptor.

7.2.2.2.8 RSV (5)

Reserved.

7.2.2.2.9 DEXT

Descriptor Extension (1b for advanced mode).

7.2.2.2.10 RSV (6)

Reserved.

7.2.2.2.11 IDX (3)

Index into the hardware context table where this context is stored.

7.2.2.2.12 RSV (1)

7.2.2.2.13 L4LEN (8)

Layer 4 header length. If *TSE* is set in the data descriptor pointing to this context, this field must be greater than or equal to 12 and less than or equal to 255. Otherwise, this field is ignored.

7.2.2.2.14 MSS (16)

Controls the Maximum Segment Size (MMS). This specifies the maximum TCP payload segment sent per frame, not including any header or trailer. The total length of each frame (or section) sent by the TCP/UDP segmentation mechanism (excluding Ethernet CRC) as follows:

Total length is equal to:

> MACLEN + 4(if VLE set) + 4 or 8(if CMTGI is set or if also RLTTGI is set - assuming BCNTLEN is clear) + IPLEN + L4LEN + MSS + [PADLEN + 18](if ESP packet)

The one exception is the last packet of a TCP/UDP segmentation, which is typically shorter.

MSS is ignored when *DCMD.TSE* is not set.

PADLEN ranges from 0 to 3 in Tx. It is the content of the *ESP Padding Length* field that is computed when offloading ESP in cipher blocks of 16-bytes (AES-128) with respect to the following alignment formula:

[L4LEN + MSS + PADLEN + 2] modulo(4) = 0

For single send packets: IPS_ESP_LEN = PADLEN + 18.

Note: The headers lengths must meet the following:

MACLEN + IPLEN + L4LEN <= 512

The context descriptor requires valid data only in the fields used by the specific offload options. The following table lists the required valid fields according to the different offload options.

Table 7-31. Valid Field in Context vs. Required Offload

Required Offload				Valid Fields in Context								
TSE	TXSM	IXSM	IPSEC	VLAN	L4LEN	IPLEN	MACLEN	MSS	L4T	IPV4	IPsec SA Index	IPsec ESP_ LEN
1b	1b	X	0b	VLE	Yes	Yes	Yes	Yes	Yes	Yes	No	No
1b	1b	X	1b	VLE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	IPSEC_ TYPE
0b	1b	X	0b	VLE	No	Yes	Yes	No	Yes	Yes	No	No
0b	1b	X	1b	VLE	No	Yes	Yes	No	Yes	Yes	Yes	IPSEC_ TYPE
0b	0b	1b	0b	VLE	No	Yes	Yes	No	No	Yes	No	No
0b	0b	1b	1b	VLE	No	Yes	Yes	No	No	Yes	Yes	IPSEC_ TYPE
0b	0b	0b	0b	No context required unless VLE is set.								
0b	0b	0b	1b	VLE	No	Yes	Yes	No	No	Yes	Yes	IPSEC_ TYPE

7.2.2.3 Advanced Transmit Data Descriptor

Table 7-32. Advanced Tx Descriptor Read Format

0	Address[63:0]									
8	PAYLEN	POPTS	CC	IDX	STA	DCMD	DTYP	MAC	RSV	DTALEN
	63 46	45 40	39	38 36	35 32	31 24	23 20	19 18	17 16	15 0

Table 7-33. Advanced Tx descriptor write-back format

0	RSV		
8	RSV	STA	RSV
	63 36	35 32	31 0

Note: For frames that spans multiple descriptors, all fields **apart** from DCMD.EOP, DCMD.RS, DCMD.DEXT, DTALEN, Address and DTYP are valid only in the first descriptors and are ignored in the subsequent ones.

7.2.2.3.1 Address (64)

Physical address of a data buffer in host memory that contains a portion of a transmit packet.

7.2.2.3.2 DTALEN (16)

Length in bytes of data buffer at the address pointed to by this specific descriptor.

Note: If the *TCTL.PSP* bit is set, the total length of the packet transmitted, not including FCS, should be at least 17 bytes.

7.2.2.3.3 RSV (2)

Reserved.

7.2.2.3.4 MAC (2)

- Reserved (bit 0)
- 1588 (bit 1) - IEEE1588 Timestamp packet.

7.2.2.3.5 DTYP (4)

0011b is the value for this descriptor type.

7.2.2.3.6 DCMD (8)

- TSE (bit 7) - TCP/UDP Segmentation Enable
- VLE (bit 6) - VLAN Packet Enable
- DEXT (bit 5) - Descriptor Extension (1b for advanced mode)
- Reserved (bit 4)
- RS (bit 3) - Report Status
- Reserved (bit 2)
- IFCS (bit 1) - Insert FCS
- EOP (bit 0) - End Of Packet

TSE indicates a TCP/UDP segmentation request. When *TSE* is set in the first descriptor of a TCP packet, hardware must use the corresponding context descriptor in order to perform TCP segmentation. The type of segmentation applied is defined according to the *TUCMD.L4T* field in the context descriptor.

Note: It is recommended that *TCTL.PSP* be enabled when *TSE* is used since the last frame can be shorter than 60 bytes - resulting in a bad frame if *PSP* is disabled.

VLE indicates that the packet is a VLAN packet and hardware must add the VLAN Ethertype and an 802.1q VLAN tag to the packet.

DEXT must be 1b to indicate advanced descriptor format (as opposed to legacy).

RS signals hardware to report the status information. This is used by software that does in-memory checks of the transmit descriptors to determine which ones are done. For example, if software queues up 10 packets to transmit, it can set the *RS* bit in the last descriptor of the last packet. If software maintains a list of descriptors with the *RS* bit set, it can look at them to determine if all packets up to (and including) the one with the *RS* bit set have been buffered in the output FIFO. Looking at the status byte and checking the *DD* bit do this. If *DD* is set, the descriptor has been processed. Refer to the sections that follow for the layout of the status field.

Note: Descriptors with zero length transfer no data.

IFCS, when set, hardware appends the MAC FCS at the end of the packet. When cleared, software should calculate the FCS for proper CRC check. There are several cases in which software must set IFCS:

- Transmitting a short packet while padding is enabled by the *TCTL.PSP* bit.
- Checksum offload is enabled by the either *TXSM* or *IXSM* bits in the TDESC.DCMD.
- VLAN header insertion enabled by the *VLE* bit in the TDESC.DCMD.
- TCP/UDP segmentation offload enabled by *TSE* bit in the TDESC.DCMD.

EOP indicates whether this is the last buffer for an incoming packet.

7.2.2.3.7 STA (4)

- Rsv (bits 1-3) - Reserved
- DD (bit 0) - Descriptor Done

7.2.2.3.8 IDX (3)

Index into the hardware context table to indicate which context should be used for this request. If no offload is required, this field is not relevant and no context needs to be initiated before the packet is sent. See Table 7-31 for details in which packets require a context reference.

7.2.2.3.9 RSV (1)

Reserved. Set to 0.

7.2.2.3.10 POPTS (6)

- RSV (bit 5:3) - Reserved
- IPSEC (bit 2) - IPSec Offload Request
- TXSM (bit 1) - Insert L4 Checksum
- IXSM (bit 0) - Insert IP Checksum

TXSM, when set, indicates that L4 checksum should be inserted. In this case, TUCMD.L4T indicates whether the checksum is TCP, UDP, or SCTP.

When *TUCMD.TSE* is set, *TXSM* must be set to 1b.

If this bit is set, the packet should at least contain a TCP header.

IXSM, when set, indicates that IP checksum should be inserted. For IPv6 packets, this bit must be cleared.

If the *TUCMD.TSE* bit is set, and *TUCMD.IPV4* is set, *IXSM* must be set as well.

If this bit is set, the packet should at least contain an IP header.

7.2.2.3.11 PAYLEN (18)

PAYLEN indicates the size (in byte units) of the data buffer(s) in host memory for transmission. In a single send packet, PAYLEN defines the entire packet size fetched from host memory. It does not include the fields that hardware adds such as: optional VLAN tagging, Ethernet CRC or Ethernet padding. When IPsec offload is enabled, it does not include the ESP trailer added by hardware. In a large send case (regardless if it is transmitted on a single or multiple packets), PAYLEN defines the protocol payload size fetched from host memory. In TCP or UDP segmentation offload, PAYLEN defines the TCP/UDP payload size.

Note: When a packet spreads over multiple descriptors, all the descriptor fields are only valid in the first descriptor of the packet, except for *RS*, which is always checked, and EOP, which is always set at last descriptor of the series.

7.2.2.4 Transmit Descriptor Ring Structure

The transmit descriptor ring structure is shown in Figure 7-12. A pair of hardware registers maintains each transmit descriptor ring in the host memory. New descriptors are added to the queue by software by writing descriptors into the circular buffer memory region and moving the tail pointer associated with that queue. The tail pointer points to one entry beyond the last hardware owned descriptor. 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.

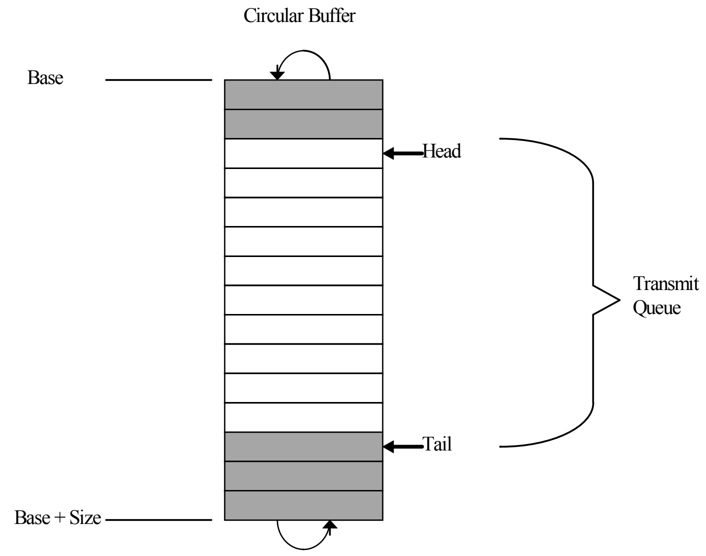


Figure 7-12. Transmit Descriptor Ring Structure

The shaded boxes in the figure represent descriptors that are not currently owned by hardware that software can modify.

The transmit descriptor ring is described by the following registers:

- Transmit Descriptor Base Address register (TDBA 0-15):

  This register indicates the start address of the descriptor ring buffer in the host memory; this 64-bit address is aligned on a 16-byte boundary and is stored in two consecutive 32-bit registers. Hardware ignores the lower four bits.
- Transmit Descriptor Length register (TDLEN 0-15):

  This register determines the number of bytes allocated to the circular buffer. This value must be zero modulo 128.
- Transmit Descriptor Head register (TDH 0-15):

  This register holds 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. Reading this register returns the value of head corresponding to descriptors already loaded in the output FIFO. This register reflects the internal head of the hardware write-back process including the descriptor in the posted write pipe and might point further ahead than the last descriptor actually written back to the memory.
- Transmit Descriptor Tail register (TDT 0-15):

  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 driver should not handle to the 82576 descriptors that describes a partial packet. Consequently, the number of descriptors used to describe a packet can not be larger than the ring size.

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 65528 (64 KB by 8) descriptors, or approximately 16 KB 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 read detect which packets had already been processed by hardware as follows:

- Read the head register to determine which packets (those logically before the head) have been transferred to the on-chip FIFO or transmitted. Note that this method is not recommended as races between the internal update of the head register and the actual write-back of descriptors might occur.
- Read the value of the head as stored at the address pointed by the TDBAH/TDBAL pair.
- Track the *DD* bits in the descriptor ring.

All the registers controlling the descriptor rings behavior should be set before transmit is enabled, apart from the tail registers which are used during the regular flow of data.

Note: Software can determine if a packet has been sent by either of three methods: setting the *RS* bit in the transmit descriptor command field or by performing a PIO read of the transmit head register, or by reading the head value written by the 82576 to the address pointed by the TDWBAL and TDWBAH registers (see Section 7.2.3 for details).

Checking the transmit descriptor *DD* bit or head value in memory eliminates a potential race condition. All descriptor data is written to the I/O bus prior to incrementing the head register, but a read of the head register could pass the data write in systems performing I/O write buffering. Updates to transmit descriptors use the same I/O write path and follow all data writes. Consequently, they are not subject to the race.

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.

7.2.2.5 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 has been increased (from 8 to 64) and the fetch and write-back algorithms modified.

When there is an on-chip descriptor buffer empty, a fetch happens as soon as any descriptors are made available (host writes to the tail pointer). If several on-chip descriptor queues are in this situation at the same time, the highest indexed queue must be served first and so forth, down to the lowest indexed queue.

A queue is considered empty for the transmit descriptor fetch algorithm as long as:

- There is still not at least one complete packet (single or large send) in its corresponding internal queue.
- There is no descriptor already in its way from system memory to the internal cache.
- The internal corresponding internal descriptor cache is not full.

Each time a descriptor fetch request is sent for an empty queue, the maximum available number of descriptor is requested, regardless of cache alignment issues.

When the on-chip buffer is nearly empty (TXDCTL[n].PTHRESH), a prefetch is performed each time enough valid descriptors (TXDCTL[n].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). If several on-chip descriptor queues are in this situation at the same time, then start from the more starved queue, and among those equally starved, start from the highest indexed queue, as before.

Note: The starvation level of a queue corresponds to the number of descriptors above the prefetch threshold that are already in the internal queue. The queue is more starved if there a less decorators in the internal queue. Comparing starvation level might be done roughly, not at the descriptor level of resolution.

When the number of descriptors in host memory is greater than the available on-chip descriptor storage, the 82576 might elect to perform a fetch that is not a multiple of cache-line size. Hardware performs this non-aligned fetch if doing so results in the next descriptor fetch being aligned on a cache-line boundary. This enables the descriptor fetch mechanism to be more efficient in the cases where it has fallen behind software.

Note: The 82576 NEVER fetches descriptors beyond the descriptor tail pointer.

7.2.2.6 Transmit Descriptor Write-Back

The descriptor write-back policy for transmit descriptors is similar to that for receive descriptors when the TXDCTL[n].WTHRESH value is not 0b. In this case, all descriptors are written back regardless of the value of their *RS* bit.

When the TXDCTL[n].WTHRESH value is 0b, since transmit descriptor write-backs do not happen for every descriptor (controlled by *RS* in the transmit descriptor), only descriptors that have *RS* bit set are written back.

Any descriptor write-back includes the full 16 bytes of the descriptor.

Since the benefit of delaying and then bursting transmit descriptor write-backs is small at best, it is likely that the threshold is 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 cases:

- TXDCTL[n].WTHRESH = 0b and a descriptor which has *RS* set is ready to be written back
- The corresponding EITR counter has reached zero
- TXDCTL[n].WTHRESH > 0b and TXDCTL[n].WTHRESH descriptors have accumulated

For the first condition, write-backs are immediate. This is the default operation and is backward compatible with previous device implementations.

The other two conditions are only valid if descriptor bursting is enabled (Section 8.12.13). In the second condition, the EITR counter 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[n].WTHRESH descriptors are ready for write-back, the write-back is performed.

An additional mode in which transmit descriptors are not written back at all and the head pointer of the descriptor ring is written instead as described in Section 7.2.3.

7.2.3 Tx Completions Head Write-Back

In legacy hardware, transmit requests are completed by writing the *DD* bit to the transmit descriptor ring. This causes cache thrash since both the software device driver and hardware are writing to the descriptor ring in host memory. Instead of writing the *DD* bits to signal that a transmit request completed, hardware can write the contents of the descriptor queue head to host memory. The software device driver reads that memory location to determine which transmit requests are complete. In order to improve the performance of this feature, the software device driver needs to program DCA registers to configure which CPU is processing each TX queue.

7.2.3.1 Description

The head counter is reflected in a memory location that is allocated by software, for each queue.

Head write-back occurs if *TDWBAL#.Head_WB_En* is set for this queue and the *RS* bit is set in the Tx descriptor, following corresponding data upload into packet buffer. If the head write-back feature is enabled, the 82576 ignores WTRESH and takes in account only descriptors with the *RS* bit set (as if the WTRESH was set to 0b). In addition, the head write-back occurs upon EITR expiration for queues where the *WB_on_EITR* field in TDWBAL is set.

The software device driver has control on this feature through Tx queue 0-15 head write-back address, low and high (thus allowing 64-bit address). See in Section 8.12.8 and Section 8.12.9.

The low register's LSB hold the control bits.

- The *Head_WB_En* bit enables activation of tail write-back. In this case, no descriptor write-back is executed.
- The 30 upper bits of this register hold the lowest 32 bits of the head write-back address, assuming that the two last bits are zero.

The high register holds the high part of the 64-bit address.

Note: Hardware writes a full Dword when writing this value, so software should reserve enough space for each head value and make sure the TDBAL value is Dword aligned.

If software enables Head Write-Back, it must also disable PCI Express Relaxed Ordering on the write-back transactions. This is done by disabling bit 11 in the TXCTL register for each active transmit queue. See Section 8.13.2.

The 82576 might update the Head with values that are larger then the last Head pointer which holds a descriptor with RS bit set, but still the value will always point to a free descriptor (descriptor that are not owned by the the 82576 anymore).

7.2.4 TCP/UDP Segmentation

Hardware TCP segmentation is one of the offloading options supported by the Windows* and Linux* TCP/IP stack. This is often referred to as TCP Segmentation Offloading or TSO. This feature enables the TCP/IP stack to pass to the network device driver a message to be transmitted that is bigger than the Maximum Transmission Unit (MTU) of medium. It is then the responsibility of the software device driver and hardware to divide 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 are constant for all packets associated with the TCP message.

The 82576 supports also UDP segmentation for embedded applications, although this offload is not supported by the regular Windows* and Linux* stacks. Any reference in this section to TCP segmentation, should be considered as referring to both TCP and UDP segmentation.

Padding (TCTL.PSP) must be enabled in TCP segmentation mode, since the last frame might be shorter than 60 bytes, resulting in a bad frame if *PSP* is disabled.

The offloading of these mechanisms to the software device driver and the 82576 save significant CPU cycles. Note that the software device driver shares the additional tasks to support these options.

7.2.4.1 Assumptions

The following assumptions apply to the TCP segmentation implementation in the 82576:

- The *RS* bit operation is not changed.
- Interrupts are set after data in buffers pointed to by individual descriptors is transferred (DMA'd) to hardware.

7.2.4.2 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 it 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 software device driver returns ownership of the packet to the Network Operating System (NOS) when hardware has completed the DMA transfer of the frame (indicated by an interrupt).

The transmission process for the 82576 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.

Hardware DMA's (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 and field updates including IPv4, IPV6, and TCP/UDP checksum generation
- The software device driver returns ownership of the block of data to the NOS when hardware has completed the DMA transfer of the entire data block (indicated by an interrupt).

7.2.4.2.1 TCP Segmentation Data Fetch Control

To perform TCP Segmentation in the 82576, the DMA must be able to fit at least one packet of the segmented payload into available space in the on-chip Packet Buffer. The DMA does various comparisons between the remaining payload and the Packet Buffer available space, fetching additional payload and sending additional packets as space permits.

In order to enable interleaving between descriptor queues at the Ethernet frame resolution inside TSO requests. For doing so, the frame header pointed by the so called header descriptors are reread from system memory by hardware for every LSO segment again, storing in an internal cache only the header's descriptors instead of the header's content.

In the aim to limit the internal cache dimensions, software is required to spread the header on maximum 4 descriptors, while still allowed to mix header and data in the last header buffer. This limitation stands for up to Layer4 header included, and for IPv4 or IPv6 indifferently.

7.2.4.2.2 TCP Segmentation Write-Back Modes

As the TCP segmentation mode uses the buffers that contains the header of the packet multiple time, there are some limitation on the usage of the different combination of writeback and buffer release methods in order to guarantee the header buffers availability until the entire packet is processed. These limitations are described in the table below.

Table 7-34. Write Back options for large send

WTHRESH	RS	HEAD Write Back Enable	Hardware Behavior	Software Expected Behavior for TSO packets.
0	Set in EOP descriptors only	Disable	Hardware writes back descriptors with RS bit set one at a time.	Software can retake ownership of all descriptors up to last descriptor with DD bit set.
0	Set in any descriptors	Disable	Hardware writes back descriptors with RS bit set one at a time.	Software can retake ownership of entire packets (EOP bit set) up to last descriptor with DD bit set.
0	Not set at all	Disable	Hardware does not write back any descriptor (since *RS* bit is not set)	Software should poll the TDH register. The TDH register reflects the last descriptor that software can take ownership of.[1]
>0	Don't care	Disable	Hardware writes back all the descriptors in bursts and set all the *DD* bits.	Software can retake ownership of entire packets up to last descriptor with both DD and EOP bits set.
Don't care	Not set at all	Enable	Hardware writes back the head pointer only at EITR expire event reflecting the last descriptor that software can take ownership of.	Software may poll the TDH register or use the head value written back at EITR expire event. The TDH register reflects the last descriptor that software can take ownership of.
Don't care	Set in EOP descriptors only	Enable	Hardware writes back the Head pointer per each descriptor with *RS* bit set.[2]	Software can retake ownership of all descriptors up to the descriptor pointed by the head pointer read from system memory (by interrupt or polling).
Don't care	Set in any descriptors	Enable	Hardware writes back the Head pointer per each descriptor with *RS* bit set.	This mode is illegal since software won't access the descriptor, it cannot tell when the pointer passed the EOP descriptor.

[1] Note that polling of the TDH register is a valid method only when the RS bit is never set, otherwise race conditions between software and hardware accesses to the descriptor ring can occur.
[2] At EITR expire event, the Hardware writes back the head pointer reflecting the last descriptor that software can take ownership of.

7.2.4.3 TCP Segmentation Performance

Performance improvements for a hardware implementation of TCP Segmentation off-load include:

- The stack does not need to partition the block to fit the MTU size, saving CPU cycles.
- The stack only computes one Ethernet, IP, and TCP header per segment, saving CPU cycles.
- The Stack interfaces with the device driver only once per block transfer, instead of once per frame.
- Larger PCI bursts are used which improves bus efficiency (such as lowering transaction overhead).
- Interrupts are easily reduced to one per TCP message instead of one per packet.

- Fewer I/O accesses are required to command the hardware.

7.2.4.4 Packet Format

Typical TCP/IP transmit window size is 8760 bytes (about 6 full size frames). Today the average size on corporate Intranets is 12-14KB, and normally the maximum window size allowed is 64KB (unless Windows Scaling - RFC 1323 is specified). A TCP message can be as large as 256 KB and is generally fragmented across multiple pages in host memory. The 82576 partitions the data packet into standard Ethernet frames prior to transmission. The 82576 supports calculating the Ethernet, IP, TCP, and UDP headers, including checksum, on a frame-by-frame basis.

Frame formats supported by the 82576 include:

Table 7-35. TCP/IP Packet Format Sent by Host

Pseudo Header			Data
Ethernet	IPv4/IPv6	TCP/UDP	DATA (full TCP message)

Table 7-36. CP/IP Packet Format Sent by the 82576

Pseudo Header (updated)	Data (first MSS)	FCS	...	Pseudo Header (updated)	Data (Next MSS)	FCS	...

- Ethernet 802.3
- IEEE 802.1Q VLAN (Ethernet 802.3ac)
- Ethernet Type 2
- Ethernet SNAP
- IPv4 headers with options
- IPv4 headers without options with one AH/ESP IPsec header
- IPv6 headers with extensions
- TCP with options
- UDP with options.

VLAN tag insertion might be handled by hardware

Note: 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 82576 has the ability to segment UDP traffic (in addition to TCP traffic), however, because UDP packets are generally fragmented at the IP layer, the 82576's “TCP Segmentation” feature is not normally conducive to handling UDP traffic.

7.2.4.5 TCP/UDP Segmentation Indication

Software indicates a TCP/UDP Segmentation transmission context to the hardware by setting up a TCP/IP Context Transmit Descriptor (see Section 7.2.2). The purpose of this descriptor is to provide information to the hardware to be used during the TCP segmentation off-load process.

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

The TCP Segmentation prototype header is taken from the packet data itself. Software must identity the type of packet that is being sent (IPv4/IPv6, TCP/UDP, other), calculate appropriate checksum off loading values for the desired checksum, and calculate the length of the header which is pre-appended. The header might be up to 240 bytes in length.

Once the TCP Segmentation context has been set, the next descriptor provides the initial data to transfer. This first descriptor(s) must point to a packet of the type indicated. Furthermore, the data it points to might 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.

IP Header

For IPv4 headers:

- *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

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 sum as follows (there is a more detailed discussion of this is Section 7.2.4.6):

Table 7-37. TCP Partial Pseudo-Header Sum for IPv4

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

Table 7-38. TCP Partial Pseudo-Header Sum for IPv6

IPv6 Source Address	
IPv6 Final Destination Address	
Zero	
Zero	Next Header

UDP Header

- Checksum should be set as in TCP header, above

The following sections describe the updating process performed by the hardware for each frame sent using the TCP Segmentation capability.

7.2.4.6 Transmit Checksum Offloading with TCP/UD Segmentation

The 82576 supports checksum off-loading as a component of the TCP Segmentation off-load feature and as a standalone capability. Section 7.2.5 describes the interface for controlling the checksum off-loading feature. This section describes the feature as it relates to TCP Segmentation.

The 82576 supports IP and TCP header options in the checksum computation for packets that are derived from the TCP Segmentation feature.

Note: The 82576 is capable of computing one level of IP header checksum and one TCP/UDP header and payload checksum. In case of multiple IP headers, the driver needs to compute all but one IP header checksum. The 82576 calculates check sums on the fly on a frame-by-frame basis and inserts the result in the IP/TCP/UDP headers of each frame.

TCP and UDP checksum are a result of performing the checksum on all bytes of the payload and the pseudo header.

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

- IPv4 checksum
- TCP checksum

Each packet that is sent via the TCP segmentation off-load feature optionally includes the IPv4 checksum and either the TCP checksum.

All checksum calculations use a 16-bit wide one's complement checksum. The checksum word is calculated on the outgoing data.

Table 7-39. Supported Transmit Checksum Capabilities

Packet Type	Hardware IP Checksum Calculation	Hardware TCP/UDP Checksum Calculation
IP v4 packets	Yes	Yes
IP v6 packets (no I checksum in Ipv6)	NA	Yes
Packet is greater than 1552 bytes; (LPE=1b)	Yes	Yes
Packet has 802.3ac tag	Yes	Yes
Packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet has TCP or UDP options	Yes	Yes
IP header’s protocol field contains a protocol # other than TCP or UDP.	Yes	No

The table below summarizes the conditions of when checksum off loading can/should be calculated.

Table 7-40. Conditions for Checksum Off Loading

Packet Type	IPv4	TCP/UDP	Reason
Non TSO	Yes	No	IP Raw packet (non TCP/UDP protocol)
	Yes	Yes	TCP segment or UDP datagram with checksum off-load
	No	No	Non-IP packet or checksum not offloaded
TSO	Yes	Yes	For TSO, checksum off-load must be done

7.2.4.7 IP/TCP/UDP Header Updating

IP/TCP or IP/UDP header is updated for each outgoing frame based on the IP/TCP header prototype which hardware DMA's from the first descriptor(s). 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 what fields are modified by hardware during the TCP Segmentation process by the 82576.

Note: Software must make PAYLEN and HDRLEN value of Context descriptors correct. Otherwise, the failure of Large Send due to either under-run or over-run might cause hardware to send bad packets or even cause TX hardware to hang. The indication of Large Send failure can be checked in the TSCTFC statistic register.

7.2.4.7.1 TCP/IP/UDP Header for the First Frames

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 + MACLEN + IPLEN + L4LEN - 14

Ipv4 Header

- IP Total Length = MSS + L4LEN + IPLEN
- IP Checksum

Ipv6 Header

- Payload Length = MSS + L4LEN + IPV6_HDR_extension[1]

TCP Header

- Sequence Number: The value is the Sequence Number of the first TCP byte in this frame.
- The flag values of the first frame are set by ANDing the flag word in the pseudo header with the DTXTCPFLGL.TCP_flg_first_seg. The default value of the DTXTCPFLGL.TCP_flg_first_seg are set so that if the FIN flag and the PSH flag are cleared in the first frame.
- TCP Checksum

7.2.4.7.2 TCP/IP 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:

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 + MACLEN + IPLEN + L4LEN - 14

Ipv4 Header

- IP Identification: incremented from last value (wrap around)
- IP Total Length = MSS + L4LEN + IPLEN
- IP Checksum

Ipv6 Header

- Payload Length = MSS + L4LEN + IPV6_HDR_extension[2]

TCP Header

[1] IPV6_HDR_extension is calculated as IPLEN - 40 bytes.
[2] IPV6_HDR_extension is calculated as IPLEN - 40 bytes.

- 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.
- The flag values of the subsequent frames are set by ANDing the flag word in the pseudo header with the DTXTCPFLGL.TCP_Flg_mid_seg. The default value of the DTXTCPFLGL.TCP_Flg_mid_seg are set so that if the FIN flag and the PSH flag are cleared in these frames.
- TCP Checksum

7.2.4.7.3 TCP / IP Header for the Last Frame

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

Last frame payload bytes = PAYLEN - (N * MSS)

MAC Header (for SNAP Packets)

- Type/Len field = Last frame payload bytes + MACLEN + IPLEN + L4LEN - 14

Ipv4 Header

- IP Total Length = last frame payload bytes + L4LEN + IPLEN
- IP Identification: incremented from last value (wrap around based on 16 bit-width)
- IP Checksum

Ipv6 Header

- Payload Length = last frame payload bytes + L4LEN + IPV6_HDR_extension[2]

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.
- The flag values of the last frames are set by ANDing the flag word in the pseudo header with the DTXTCPFLGH.TCP_Flg_lst_seg. The default value of the DTXTCPFLGH.TCP_Flg_lst_seg are set so that if the FIN flag and the PSH flag are set in the last frame.
- TCP Checksum

7.2.4.8 IP / TCP / UDP Checksum Offloading

The 82576 performs checksum off loading as part of the TCP segmentation off-load feature.

These specific checksum are supported under TCP segmentation:

- IPv4 checksum
- TCP checksum

See Section 7.2.5 for description of checksum off loading of a single-send packet.

7.2.4.9 Data Flow

The flow used by the 82576 to do a TCP segmentation is as follow:

1. Get a descriptor with a request for a TSO off-load of a TCP packet.
2. First Segment processing:
    a. Fetch all the buffers containing the header as calculated by the MACLEN, IPLEN & L4LEN fields. Save the addresses and lengths of the buffers containing the header (up to 4 buffers). The header content is not saved.

   b. Fetch data up to the MSS from subsequent buffers & calculate the adequate checksum(s).
   c. Update the Header accordingly and update internal state of the packet (next data to fetch and IP SN).
   d. Send the packet to the network.
   e. If total packet was sent, go to step 4. else continue.
3. Next segments
   a. Wait for next arbitration of this queue.
   b. Fetch all the buffers containing the header from the saved addresses. Subsequent reads of the header might be done with a no snoop attribute.
   c. Fetch data up to the MSS or end of packet form subsequent buffers & calculate the adequate checksum(s.
   d. Update the Header accordingly and update internal state of the packet (next data to fetch and IP SN).
   e. If total packet was sent, request is done, else restart from step 3.
4. Release all buffers (update head pointer).

Note: Descriptors are fetched in a parallel process according to the consumption of the buffers.

7.2.5 Checksum Offloading in Non-Segmentation Mode

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

Before taking advantage of the 82576's enhanced checksum off-load capability, a checksum context must be initialized. For the normal transmit checksum off-load feature this is performed by providing the device with a TCP/IP Context Descriptor with TUCMD.TSE=0b. Setting TSE=0b indicates that the normal checksum context is being set, as opposed to the segmentation context. For additional details on contexts, refer to Section 7.2.2.4.

Note: Enabling the checksum off loading capability without first initializing the appropriate checksum context leads to unpredictable results. CRC appending (CMD.IFCS) must be enabled in TCP/IP checksum mode, since CRC must be inserted by hardware after the checksum have been calculated.

As mentioned in Section 7.2.2, Transmit Descriptors, 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 performance can be gained by only changing the context on an as needed basis or electing to use the off-load feature only for a particular traffic type, thereby avoiding all context descriptors except for the initial one.

Each checksum operates independently. Insertion of the IP and TCP checksum for each packet are enabled through the Transmit Data Descriptor POPTS.TSXM and POPTS.IXSM fields, respectively.

7.2.5.1 IP Checksum

Three fields in the Transmit Context Descriptor set the context of the IP checksum off loading feature:

- TUCMD.IPv4
- IPLEN
- MACLEN

TUCMD.IPv4=1b specifies that the packet type for this context is IPv4, and that the IP header checksum should be inserted. TUCMD.IPv4=0b indicates that the packet type is IPv6 (or some other protocol) and that the IP header checksum should not be inserted.

MACLEN specifies the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, the start of the IP header. The minimal allowed value for this field is 12. Note that the maximum value for this field is 127. This is adequate for typical applications.

Note: The MACLEN+IPLEN value needs to be less than the total DMA length for a packet. If this is not the case, the results are unpredictable.

IPLEN specifies the IP header length. Maximum allowed value for this field is 511 Bytes.

MACLEN+IPLEN specify where the IP checksum should stop. This is limited to the first 127+511 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 result is unpredictable.

Note: For IPsec packet offloaded by hardware in Tx, it is assumed that IPLEN provided by software in the Tx context descriptor is the sum of the IP header length with the IPsec header length. Thus For the IPv4 header checksum off-load, hardware could no more rely on the *IPLEN* field provided by software in the Tx context descriptor, but should rely on the fact that no IPv4 options are present in the packet. Consequently, for IPsec off-load packets hardware computes IP header checksum over always a fixed amount of 20-bytes.

The 16-bit IPv4 Header Checksum is placed at the two bytes starting at MACLEN+10.

As mentioned in Section 7.2.2.2, Transmit Contexts, 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 performance can be gained by only changing the context on an as needed basis or electing to use the off-load feature only for a particular traffic type, thereby avoiding all context descriptors except for the initial one.

7.2.5.2 TCP Checksum

Three fields in the Transmit Context Descriptor set the context of the TCP checksum off loading feature:

- MACLEN
- IPLEN
- TUCMD.L4T

TUCMD.L4T=1b specifies that the packet type is TCP, and that the 16-bit TCP header checksum should be inserted at byte offset MACLEN+IPLEN+16. TUCMD.L4T=0b indicates that the packet is UDP and that the 16-bit checksum should be inserted starting at byte offset MACLEN+IPLEN+6.

IPLEN+MACLEN specifies the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, the start of the TCP header. The minimal allowed value for this sum is 32/42 for UDP or TCP respectively.

Note: The IPLEN+MACLEN+L4LEN value needs to be less than the total DMA length for a packet. If this is not the case, the results are unpredictable.

The TCP/UDP checksum always continues to the last byte of the DMA data.

Note: For non-TSO, software still needs to calculate a full checksum for the TCP/UDP pseudo-header. This checksum of the pseudo-header should be placed in the packet data buffer at the appropriate offset for the checksum calculation.

7.2.5.3 SCTP CRC Offloading

For SCTP packets, a CRC32 checksum offload is provided.

Three fields in the Transmit Context Descriptor set the context of the STCP checksum off loading feature:

- MACLEN
- IPLEN
- TUCMD.L4T

TUCMD.L4T=10b specifies that the packet type is SCTP, and that the 32-bit STCP CRC should be inserted at byte offset MACLEN+IPLEN+8.

IPLEN+MACLEN specifies the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, the start of the STCP header. The minimal allowed value for this sum is 26.

The SCTP CRC calculation always continues to the last byte of the DMA data.

The SCTP total L3 payload size (PAYLEN - IPLEN - MACLEN) should be a multiple of 4 bytes (SCTP padding not supported).

Note: TSO is not available for SCTP packets.

7.2.5.4 Checksum Supported Per Packet Types

The following table summarizes which checksum is supported per packet type.

Note: TSO is not supported for packet types for which IP checksum & TCP checksum can not be calculated.

Table 7-41. Checksum Per Packet Type

Packet Type	Hardware IP Checksum Calculation	Hardware TCP/UDP/SCTP 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 (w len 0b) • Routing (w len >0b) • Fragment • Home option • Security Option (AH/ESP)	 No (n/a) No (n/a) No (n/a) No (n/a) No (n/a) No (n/a) Yes[1]	 Yes Yes Yes No No No Yes[1]
Ipv4 tunnels: • Ipv4 packet in an Ipv4 tunnel • Ipv6 packet in an Ipv4 tunnel	 Either IP or TCP/SCTP [2] Either IP or TCP/SCTP [2]	 Either IP or TCP/SCTP [2] Either IP or TCP/SCTP [2]
Ipv6 tunnels: • Ipv4 packet in an Ipv6 tunnel • Ipv6 packet in an Ipv6 tunnel	 No No	 Yes Yes
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 and no IPSec header (IP header is longer than 20 bytes)	Yes	Yes

Table 7-41. Checksum Per Packet Type (Continued)

Ipv4 Packet has IPSec Header without IP options	Yes[1]	Yes[1]
Packet has TCP or UDP options	Yes	Yes
IP header's protocol field contains protocol # other than TCP or UDP.	Yes	No

[1] Only offloaded flows

[2] For the tunneled case, the driver might do only the TCP checksum or Ipv4 checksum. If TCP checksum is desired, the driver should define the IP header length as the combined length of both IP headers in the packet. If an IPv4 checksum is required, the IP header length should be set to the Ipv4 header length.

7.2.6 Multiple Transmit Queues

The number of transmit queues is increased to 16, to match the expected number of processors on most server platforms and to support the new virtualization mode.

If there are more CPUs than queues, then one queue might be used to service more than one CPU.

For transmission process, each thread might set a queue in the host memory of the CPU it is tied to.

7.2.6.1 Bandwidth Allocation to Virtual Machines / Transmit Queues

When operated in either VMDq2 or SR-IOV mode, the 82576 has the ability to control the Tx bandwidth used by each Virtual Machine (VM). Since in these virtualization modes each Tx Queue is owned by a separate VM (or a separate set of VMs), bandwidth allocation to VMs is performed by assigning bandwidth shares to Tx Queues. A rate-controller is internally associated to a Tx Queue to maintain its allocated bandwidth share.

The bandwidth share represents the minimum percentage of the link's bandwidth that is guaranteed to be granted to the VM. Bandwidth unused by a VM is re-distributed among others according to their relative bandwidth shares.

If the PCIe bandwidth available for transmission get below half of the link's bandwidth, the bandwidth allocation to VMs scheme will degenerate into a scheme close to a packet-based round-robin arbitration between the VMs.

If the link is operated at 10Mbps, bandwidth allocation to VMs must be disabled as in non-virtualized contexts, and Tx Queues are served in a packet-based round-robin manner.

A VM can be operated in a "Bandwidth Takeover" mode, where it takes over for itself all bandwidth left unused by others. When several VMs are operated in this mode, unused bandwidth left by others is equally distributed among them, in a packet-based round-robin manner.

The bandwidth share scheme is configured by the following set of registers:

- VMBACS, to control the general operation of the bandwidth allocation to VMs feature.
- VMBAMMW, to set the maximum amount of Tx payload compensation a VM can accumulate in case it temporarily does not use its allocated bandwidth.
- VMBASEL, to select the VM / Tx Queue for which a bandwidth share is configured via the VMBAC register.
- VMBAC, to set the minimum rate allocated to a VM.

7.3 Interrupts

7.3.1 Mapping of Interrupt Causes

The 82576 supports the following interrupt modes:

- PCI legacy interrupts or MSI - selected when GPIE.Multiple_MSIX is 0b
- MSI-X in non-IOV mode - selected when GPIE.Multiple_MSIX is 1b and the *VFE* bit in PCIe SR-IOV control register is cleared.
- MSI-X in IOV mode - selected when GPIE.Multiple_MSIX is 1b and the *VFE* bit in PCIe SR-IOV control register is set.

Note: If only one MSI-X vector is allocated by the operating system, then the driver might use the non MSI-X mapping method even in MSI-X mode.

Mapping of interrupts causes is different in each of the above modes and is described below.

7.3.1.1 Legacy and MSI Interrupt Modes

In legacy and MSI modes, an interrupt cause is reflected by setting a bit in the EICR register. This section describes the mapping of interrupt causes (a specific Rx queue event or a LSC event) to bits in the EICR.

Mapping of queue-related causes is accomplished through the IVAR register. Each possible queue interrupt cause (each Rx or Tx queue) is allocated an entry in the IVAR, and each entry in the IVAR identifies one bit in the EICR register among the bits allocated to queue interrupt causes. It is possible to map multiple interrupt causes into the same *EICR* bit.

In this mode, causes can be mapped to the first 16 bits of the EICR register.

Interrupt causes related to non-queue causes are mapped into the ICR legacy register; each cause is allocated a separate bit. The sum of all causes is reflected in the *Other* bit in EICR. Figure 7-13 below describes the allocation process.

The following configuration and parameters are involved:

- The IVAR[7:0] entries map 16 Tx queues and 16 Rx queues into EICR[15:0] bits
- The IVAR_MISC that maps non-queue causes is not used
- The *EICR[30]* bit is allocated to the TCP timer interrupt cause.
- The *EICR[31]* bit is allocated to the other interrupt causes summarized in the ICR reg.
- A single interrupt vector is provided.

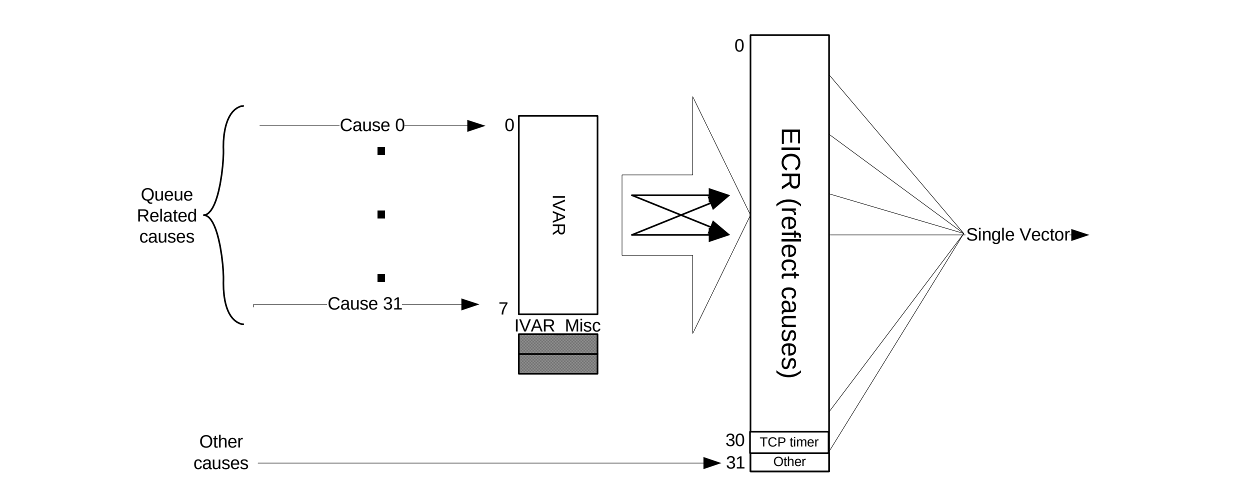


Figure 7-13. Cause Mapping in Legacy Mode

The Table below maps the different interrupt causes into the IVAR registers.

Table 7-42. Cause Allocation in the IVAR Registers - MSI and Legacy Mode

Interrupt	Entry	Description
Rx_i	i*4 (i= 0..7)	Receive queues i - Associates an interrupt occurring in the Rx queues i with a corresponding bit in the EICR register.
Tx_i	i*4+1 (i= 0..7)	Transmit queues i- Associates an interrupt occurring in the Tx queues I with a corresponding bit in the EICR register.
Rx_i	(i-8)*4+2 (i= 8..15)	Receive queues i - Associates an interrupt occurring in the Rx queues i with a corresponding bit in the EICR register.
Tx_i	(i-8)*4+3 (i= 8..15)	Transmit queues i- Associates an interrupt occurring in the Tx queues I with a corresponding bit in the EICR register.

7.3.1.2 MSI-X Mode - Non-IOV Mode

In a non Single Root - IOV setup (SR-IOV capability is not exposed in the PCIe configuration space), the 82576 can request up to 25 Vectors.

In MSI-X mode, an interrupt cause is mapped into an MSI-X vector. This section describes the mapping of interrupt causes (a specific Rx queue event or other events) to MSI-X vectors.

Mapping is accomplished through the IVAR register. Each possible cause for an interrupt is allocated an entry in the IVAR, and each entry in the IVAR identifies one MSI-X vector. It is possible to map multiple interrupt causes into the MSI-X vector.

The EICR also reflects interrupt vectors. The EICR bits allocated for queue causes reflect the MSI-X vector (bit 2 is set when MSI-X vector 2 is used). Interrupt causes related to non-queue causes are mapped into the ICR (as in the legacy case). The MSI-X vector for all such causes is reflected in the EICR.

The following configuration and parameters are involved:

- The IVAR[7:0] entries map 16 Tx queues, 16 Rx queues, a TCP timer, and other events to up to 23 interrupt vectors
- The IVAR_MISC register maps a TCP timer and other events to 2 MSI-X vectors

Figure 7-14 describes the allocation process.

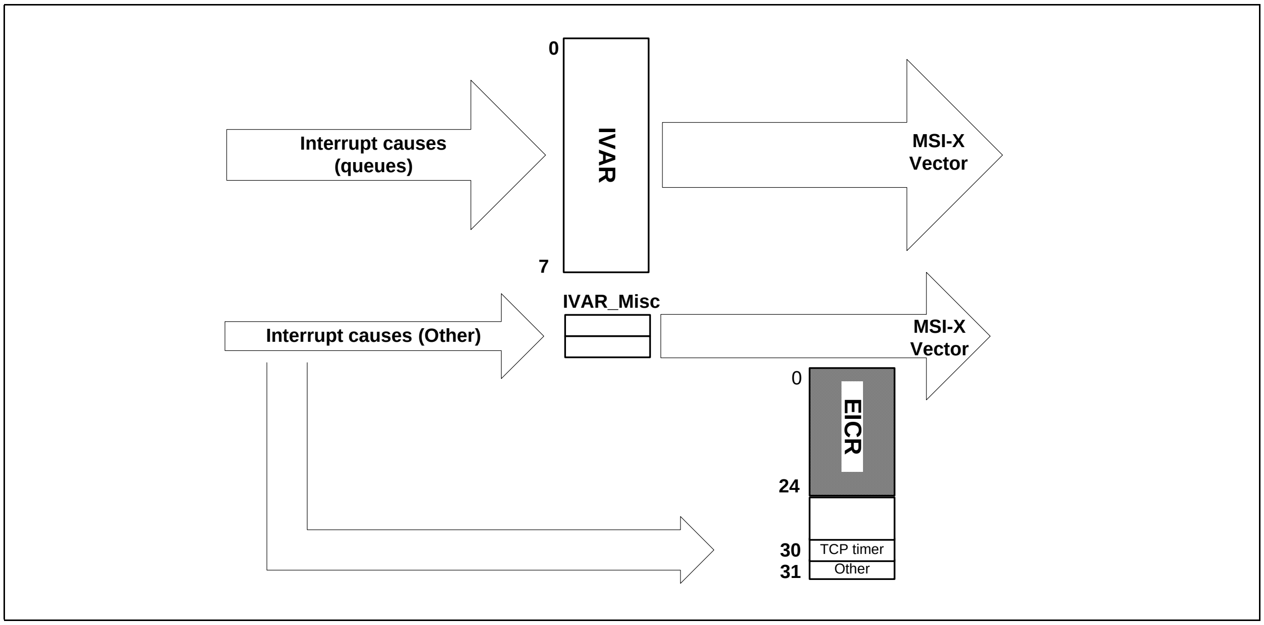


Figure 7-14. Cause Mapping in MSI-X Mode

Table 7-43 below defines which interrupt cause is represented by each entry in the MSI-X Allocation registers.

In non SR-IOV mode, the software has access to 34 mapping entries to map each cause to one of the 25 MSI-x vectors.

Table 7-43. Cause Allocation in the IVAR Registers - Non-IOV Mode

Interrupt	Entry	Description
Rx_i	i*4 (i= 0..7)	Receive queues i - Associates an interrupt occurring in the Rx queues i with a corresponding entry in the MSI-X Allocation registers.
Tx_i	i*4+1 (i= 0..7)	Transmit queues i- Associates an interrupt occurring in the Tx queues I with a corresponding entry in the MSI-X Allocation registers.
Rx_i	(i-8)*4+2 (i= 8..15)	Receive queues i - Associates an interrupt occurring in the Rx queues i with a corresponding entry in the MSI-X Allocation registers.
Tx_i	(i-8)*4+3 (i= 8..15)	Transmit queues i- Associates an interrupt occurring in the Tx queues I with a corresponding entry in the MSI-X Allocation registers.
TCP timer	32	TCP Timer - Associates an interrupt issued by the TCP timer with a corresponding entry in the MSI-X Allocation registers
Other cause	33	Other causes - Associates an interrupt issued by the “other causes” with a corresponding entry in the MSI-X Allocation registers

7.3.1.3 MSI-X Interrupts in SR-IOV Mode

Each of the VF functions in PCI-SIG SR-IOV mode is allocated 3 MSI-X vectors. The PF can request up to 10 vectors.

Interrupt allocation for the physical function (PF) is done as in the MSI-X non-IOV case. However, the PF should not assign interrupt vectors to queues not assigned to it. The IVAR_MISC register allocates non-queue interrupts as in the non-IOV case with a single change - the entry assigned to “other” causes also handles interrupt on the mailbox.

Although the PF is allocated up to 10 vectors, these vectors shares the internal interrupts with the VFs. See Section 7.3.3.1 for details of the sharing of the internal interrupts.

Each of the VFs in IOV mode is allocated separate IVAR registers (called VTIVAR), translating its queue-related interrupt causes into MSI-X vectors for this virtual function. The IVAR register has one entry per Tx or Rx queue. A VTIVAR_MISC register is provided to map the mailbox interrupt into an MSI-X vector.

The PF can allocate interrupt causes not used by the VFs to one of it's own vectors.

The EICR of each VF or of the PF reflects the status of the MSI-X vectors allocated to this function.

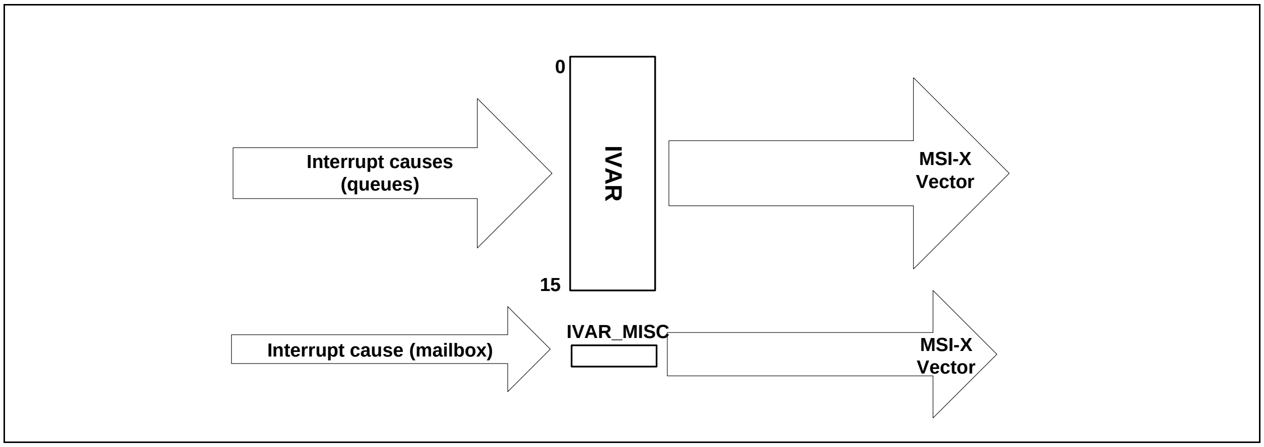


Figure 7-15. Cause Mapping of a VF in MSI-X Mode (IOV)

Table 7-44 below, defines for a given VM (not PF) which interrupt cause is represented by each entry in the MSI-X Allocation registers.

In the IOV mode the software have access to 5 mapping entries to map each cause to one out of 3 MSI-x vectors

The 3 VM vectors (per each VM) can be allocated to one or more causes (2 Q traffic interrupt, Mail Box interrupt).

Please refer to Section 7.9.2.9.1 for details of the Mail Box mechanism.

Table 7-44. Cause Allocation for a VF in the VTIVAR Registers - IOV Mode

Interrupt	Entry	Description
Rx Queue i (i=0...1)	i*2	Receive queue i - Associates an interrupt occurring in Rx queue i with a corresponding entry in the MSI-X Allocation registers.
Tx Queue i (i=0...1)	i*2+1	Transmit queue i- Associates an interrupt occurring in Tx queue i with a corresponding entry in the MSI-X Allocation registers.

7.3.2 Registers

The interrupt logic consists of the registers listed in the tables below, plus the registers associated with MSI/MSI-X signaling. The first table describes the use of the registers in legacy mode and the second one the use of the register when using the extended interrupts functionality

Table 7-45. Interrupt Registers - Legacy Mode

Register	Acronym	Function
Interrupt Cause	ICR	Records interrupt conditions.
Interrupt Cause Set	ICS	Allows software to set bits in the ICR.
Interrupt Mask Set/Read	IMS	Sets or reads bits in the interrupt mask.
Interrupt Mask Clear	IMC	Clears bits in the interrupt mask.
Interrupt Acknowledge auto-mask	IAM	Under some conditions, the content of this register is copied to the mask register following read or write of ICR.

Table 7-46. Interrupt Registers - Extended Mode

Register	Acronym	Function
Extended Interrupt Cause	EICR	Records interrupt causes from receive and transmit queues. An interrupt is signaled when unmasked bits in this register are set.
Extended Interrupt Cause Set	EICS	Allows software to set bits in the Interrupt Cause register.
Extended Interrupt Mask Set/Read	EIMS	Sets or read bits in the interrupt mask.
Extended Interrupt Mask Clear	EIMC	Clears bits in the interrupt mask.
Extended Interrupt Auto Clear	EIAC	Allows bits in the EICR to be cleared automatically following an MSI-X interrupt without a read or write of the EICR.
Extended Interrupt Acknowledge auto-mask	EIAM	This register is used to decide which masks are cleared in the extended mask register following read or write of EICR or which masks are set following a write to EICS. In MSI-X mode, this register also controls which bits in EIMC are cleared automatically following an MSI-X interrupt.
Interrupt Cause	ICR	Records interrupt conditions for special conditions - a single interrupt from all the conditions of ICR is reflected in the “other” field of the EICR.
Interrupt Cause Set	ICS	Allows software to set bits in the ICR.
Interrupt Mask Set/Read	IMS	Sets or reads bits in the other interrupt mask.
Interrupt Mask Clear	IMC	Clears bits in the Other interrupt mask.
Interrupt Acknowledge auto-mask	IAM	Under some conditions, the content of this register is copied to the mask register following read or write of ICR.
General Purpose Interrupt Enable	GPIE	Controls different behaviors of the interrupt mechanism.

7.3.2.1 Interrupt Cause Register (ICR)

7.3.2.1.1 Legacy Mode

In Legacy mode, ICR is used as the sole interrupt cause register. Upon reception of an interrupt, the interrupt handling routine can read this register in order to find out what are the causes of this interrupt.

7.3.2.1.2 Advanced Mode

In advanced mode, this register captures the interrupt causes not directly captured by the EICR. These are infrequent management interrupts and error conditions.

Note that when EICR is used in advanced mode, the RX /TX related bits in ICR should be masked.

ICR bits are cleared on register read. If GPIE.NSICR = 0b, then the clear on read occurs only if no bit is set in the IMS or at least one bit is set in the IMS and there is a true interrupt as reflected in ICR.INTA.

7.3.2.2 Interrupt Cause Set Register (ICS)

This registers allows setting the bits of ICR by software, by writing a 1b in the corresponding bits in ICS. Used usually to rearm interrupts the software didn't have time to handle in the current interrupt routine.

7.3.2.3 Interrupt Mask Set/Read Register (IMS)

An interrupt is enabled if its corresponding mask bit in this register is set to 1b, and disabled if its corresponding mask bit is set to 0b. A PCIe interrupt is generated whenever 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 Register.

Reading this register returns which bits have an interrupt mask set.

A particular interrupt might be enabled by writing a 1b to the corresponding mask bit in this register. Any bits written with a 0b are unchanged. Thus, if software desires to disable a particular interrupt condition that had been previously enabled, it must write to the Interrupt Mask Clear Register (see below), rather than writing a 0b to a bit in this register.

7.3.2.4 Interrupt Mask Clear Register (IMC)

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).

7.3.2.5 Interrupt Acknowledge Auto-mask register (IAM)

An ICR read or write has the side effect of writing the contents of this register to the mask register. If GPIE.NSICR = 0b, then the copy of this register to the mask register occurs only at least one bit is set in the mask register and there is a true interrupt as reflected in ICR.INTA.

7.3.2.6 Extended Interrupt Cause Registers (EICR)

7.3.2.6.1 MSI/INT-A Mode

This register records the interrupts causes to provide to the software information on the interrupt source.

The interrupt causes include:

1. The Receive and Transmit queues — Each queue (either Tx or Rx) can be mapped to one of the 16 interrupt causes bits (RTxQ) available in this register according to the mapping in the IVAR registers
2. Indication for the TCP timer interrupt.
3. Legacy and other indications — When any interrupt in the Interrupt Cause register is active.

Writing 1bs clears the corresponding bits in this register. Most systems have write-buffering that minimizes overhead, but this might require a read operation to guarantee that the write has been flushed from posted buffers. Reading this register auto-clears all bits.

7.3.2.6.2 MSI-X Mode

This register records the interrupt vectors currently emitted. In this mode only the first 25 bits are valid.

For all the subsequent registers, in MSI-X mode, each bit controls the behavior of one vector.

Bits in this register can be configured to auto-clear when the MSI-X interrupt message is sent, in order to minimize driver overhead when using MSI-X interrupt signaling.

7.3.2.7 Extended Interrupt Cause Set Register (EICS)

This registers allows to set the bits of EICR by software, by writing a 1b in the corresponding bits in EICS. Used usually to rearm interrupts the software didn't have time to handle in the current interrupt routine.

7.3.2.8 Extended Interrupt Mask Set and Read Register (EIMS) & Extended Interrupt Mask Clear Register (EIMC)

Interrupts appear on PCIe only if the interrupt cause bit is a one and the corresponding interrupt mask bit is a one. Software blocks assertion of an interrupt by clearing the corresponding bit in the mask register. The cause bit stores the interrupt event regardless of the state of the mask bit. Different Clear (EIMC) and set (EIMS) registers make this register more “thread safe” by avoiding a read-modify-write operation on the mask register. The mask bit is set for each bit written to a one in the set register (EIMS) and cleared for each bit written in the clear register (EIMC). Reading the set register (EIMS) returns the current mask register value.

7.3.2.9 Extended Interrupt Auto Clear Enable Register (EIAC)

Each bit in this register enables clearing of the corresponding bit in EICR following interrupt generation. When a bit is set, the corresponding bit in EICR are automatically cleared following an interrupt. This feature should only be used in MSI-X mode.

When used in conjunction with MSI-X interrupt vector, this feature allows interrupt cause recognition, and selective interrupt cause, without requiring software to read or write the EICR register; therefore, the penalty related to a PCIe read or write transaction is avoided.

The process of interrupt cause bits reset is described below in See “Clearing Interrupt Causes

7.3.2.10 Extended Interrupt Auto Mask Enable Register (EIAM)

Each bit set in this register enables clearing of the corresponding bit in the extended mask register following read or write-to-clear to EICR. It also enables setting of the corresponding bit in the extended mask register following a write-to-set to EICS.

This mode is provided in case MSI-X is not used, and therefore auto-clear through EIAC register is not available.

In MSI-X mode, the driver software might set the bits of this register to select mask bits that must be reset during interrupt processing. In this mode, each bit in this register enables clearing of the corresponding bit in EIMC following interrupt generation.

7.3.2.11 GPIE

There are a few bits in the GPIE register that define the behavior of the interrupt mechanism. The setting of these bits is different in each mode of operation. The following table describes the recommended setting of these bits in the different modes:

Table 7-47. Settings for Different Interrupt Modes

Field	Bit(s)	Initial Value	Description	INT-x / MSI + Legacy	INT-x / MSI + Extend	MSI-X Multi vector	MSI-X Single vector
NSICR	0	0b	**Non Selective Interrupt clear on read**: When set, every read of ICR clears it. When this bit is cleared, an ICR read causes it to be cleared only if an actual interrupt was asserted or IMS = 0b.	0b[1]	1b	1b	1b
Multiple_ MSIX	4	0b	**Multiple_MSIX - multiple vectors:** 0b = non-MSIX or MSI-X with 1 vector IVAR map Rx/Tx causes to 16 EICR bits, but MSIX[0] is asserted for all 1b = MSIX mode, IVAR maps Rx/Tx causes to 25 EICR bits	0b	0b	1b	0b
EIAME	30	0b	**EIAME:** When set, upon firing of an MSI-X message, mask bits set in EIAM associated with this message are cleared. Otherwise, EIAM is used only upon read or write of EICR/EICS registers.	0b	0b	1b	1b
PBA_ support	31	0b	**PBA support:** When set, setting one of the extended interrupts masks via EIMS causes the PBA bit of the associated MSI-X vector to be cleared. Otherwise, the 82576 behaves in a way supporting legacy INT-x interrupts. Should be cleared when working in INT-x or MSI mode and set in MSI-X mode.	0b	0b	1b	1b

[1] In systems where interrupt sharing is not expected, the *NSICR* bit can be set by legacy drivers also

As this register affects the way the hardware interprets write to the other interrupt control registers, it should be set the to the right mode before any access to the other registers.

7.3.3 MSI-X and Vectors

MSI-X defines a separate optional extension to basic MSI functionality. Compared to MSI, MSI-X supports a larger maximum number of vectors per function, the ability for software to control aliasing when fewer vectors are allocated than requested, plus the ability for each vector to use an independent address and data value, specified by a table that resides in Memory Space. However, most of the other characteristics of MSI-X are identical to those of MSI. For more information on MSI-X, refer to the PCI Local Bus Specification, Revision 3.0.

MSI-X maps each of the Intel® 82576 GbE Controller interrupt causes into an interrupt vector that is conveyed by the 82576 as a posted-write PCIe transaction. Mapping of an interrupt cause into an MSI-X vector is determined by system software (a device driver) through a translation table stored in the MSI-X Allocation registers. Each entry of the allocation registers defines the vector for a single interrupt cause.

There are 34 extended interrupt causes exit in the 82576:

1. 32 traffic causes — 16 Tx, 16 Rx.
2. TCP timer
3. Other causes — Summarizes legacy interrupts into one extended cause.

The way the 82576 exposes causes to the software is determined by the IOV mode. See Section 7.3.1 for details.

7.3.3.1 Usage of Spare MSI-X Vectors by Physical Function

The total number of available MSI-X vector is 34. The PF should not request vectors that may be later allocated to VFs. For example, if the driver knows that at most 6 VFs will be enabled, it can request up to 34 - 3*6 = 16 vectors. In any case, the PF can request 10 vectors, even if all the VFs are allocated. However, the number of internal interrupts is only 25, thus when VFs are enabled, the PF should release all the internal interrupt resources allocated to the VFs. The following table describes the PF vectors available according to the number of VFs enabled assuming the PF requests up to 10 vectors. The available vectors can be referenced in the IVAR registers and indicates which EITR registers are available for the PF.

Table 7-48. Internal vectors available to the PF

Enabled VFs	Available vectors
0-5	0-9
6	0-7
7	0-3
8	0

7.3.4 Interrupt Moderation

An interrupt is generated upon receiving of incoming packets, as throttled by the EITR registers (see Section 8.8.12). There is an EITR register per MSI-X vector.

In MSI-X mode, each active bit in EICR can trigger the interrupt vector it is allocated to. Following the allocation, the EITR corresponding to the MSI-X vector is tied to one or more bits in EICR.

When MSI-X is not activated, the interrupt moderation is controlled by EITR[0].

Software can use EITR to limit the rate of delivery of interrupts to the host CPU. This register provides a guaranteed inter-interrupt delay between interrupts asserted by the network controller, regardless of network traffic conditions.

The following formula converts the inter-interrupt interval value to the common 'interrupts/sec.' performance metric:

$$\text{interrupts/sec} = (2 * 10^{-6}\text{sec x interval})^{-1}$$

Note: In the 82576 the interval granularity is 2 µsec so some of the LSB of the interval are used for the low latency interrupt moderation.

For example, if the interval is programmed to 125d, the network controller guarantees the CPU is not interrupted by the network controller for at least 250 µs from the last interrupt. In this case, the maximum observable interrupt rate from the adapter should not exceed 4000 interrupts/sec.

Inversely, inter-interrupt interval value can be calculated as:

$$\text{inter-interrupt interval} = (2 * 10^{-6}\text{ sec x interrupt/sec})^{-1}$$

The optimal performance setting for this register is very system and configuration specific.

The Extended Interrupt Throttle Register should default to zero upon initialization and reset. It loads in the value programmed by the software after software initializes the device.

When software wants to force an immediate interrupt, for example after setting a bit in the EICR with the Extended Interrupt Cause Set register, the value of the Counter can be written to 0b to generate an interrupt immediately. This write should include re-writing the *Interval* field with the desired constant, as it is used to reload the Counter immediately for the next throttling interval.

The 82576 implements interrupt moderation to reduce the number of interrupts software processes. The moderation scheme is based on the EITR (Interrupt Throttle Register). Whenever an interrupt event happens, the corresponding bit in the EICR is activated. However, an interrupt message is not sent out on the PCIe interface until the EITR counter assigned to that *EICR* bit has counted down to zero. As soon as the interrupt is issued, the EITR counter is reloaded with its initial value and the process repeats again. The interrupt flow should follow the diagram below:

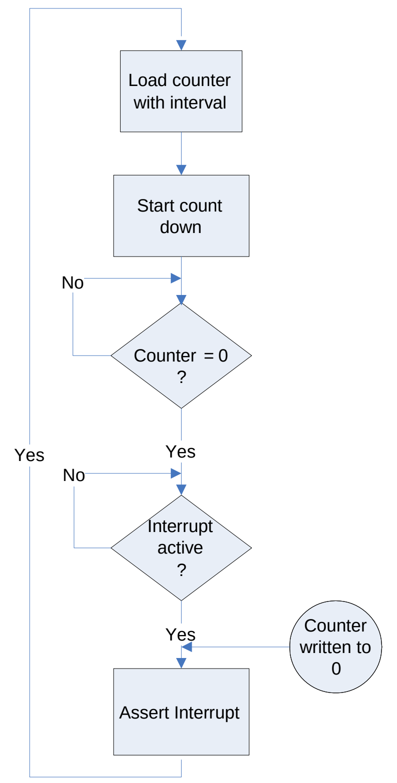


Figure 7-16. Interrupt Throttle Flow Diagram

For cases where the 82576 is connected to a small number of clients, it is desirable to fire off the interrupt as soon as possible with minimum latency. For these cases, when the EITR counter counts down to zero and no interrupt event has happened, then the EITR counter is not reset but stays at zero. Thus, the next interrupt event triggers an interrupt immediately. That scenario is illustrated as “Case B” below.

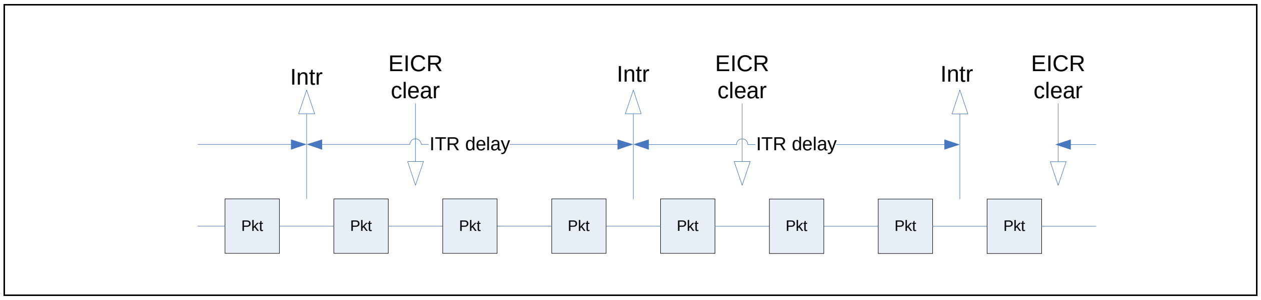


Figure 7-17. Case A: Heavy Load, Interrupts Moderated

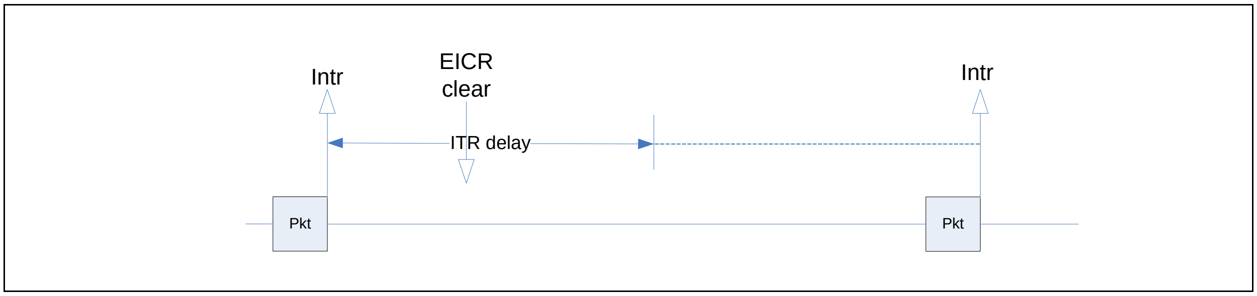


Figure 7-18. Light load, Interrupts Immediately on Packet Receive

7.3.5 Clearing Interrupt Causes

The 82576 has three methods available for to clear EICR bits: Autoclear, clear-on-write, and clear-on-read. ICR bits might only be cleared with clear-on-write or clear-on-read.

7.3.5.1 Auto-Clear

In systems that support MSI-X, the interrupt vector allows the interrupt service routine to know the interrupt cause without reading the EICR. With interrupt moderation active, software load from spurious interrupts is minimized. In this case, the software overhead of a I/O read or write can be avoided by setting appropriate EICR bits to autoclear mode by setting the corresponding bits in the Extended Interrupt Auto-clear Enable Register (EIAC).

When auto-clear is enabled for an interrupt cause, the *EICR* bit is set when a cause event mapped to this vector occurs. When the EITR Counter reaches zero, the MSI-X message is sent on PCIe. Then the *EICR* bit is cleared and enabled to be set by a new cause event. The vector in the MSI-X message signals software the cause of the interrupt to be serviced.

It is possible that in the time after the *EICR* bit is cleared and the interrupt service routine services the cause, for example checking the transmit and receive queues, that another cause event occurs that is then serviced by this ISR call, yet the *EICR* bit remains set. This results in a “spurious interrupt”. Software can detect this case, for example if there are no entries that require service in the transmit and receive queues, and exit knowing that the interrupt has been automatically cleared. The use of interrupt moderations through the EITR register limits the extra software overhead that can be caused by these spurious interrupts.

7.3.5.2 Write to Clear

In the case where the driver wishes to configure itself in MSI-X mode to not use the “auto-clear” feature, it might clear the EICR bits by writing to the EICR register. Any bits written with a 1b is cleared. Any bits written with a 0b remain unchanged.

7.3.5.3 Read to Clear

The EICR and ICR registers are cleared on a read.

Note that the driver should never do a read-to-clear of the EICR when in MSI-X mode, since this might clear interrupt cause events which are processed by a different interrupt handler (assuming multiple vectors).

7.3.6 Rate Controlled Low Latency Interrupts (LLI)

There are some types of network traffic for which latency is a critical issue. For these types of traffic, interrupt moderation hurts performance by increasing latency between the time a packet is received by hardware and the time it is handled to the host operating system. This traffic can be identified by the 5-tuple value, in conjunction with Control Bits and specific size. In addition packets with specific ethernet types, TCP flag or specific VLAN priority might generate an immediate interrupt.

Low latency interrupts shares the filters used by the queueing mechanism described in Section 7.1.1. Each of these filters, in addition to the queueing action might also indicate matching packets might generate immediate interrupt.

If a received packet matches one of these filters, hardware should interrupt immediately, overriding the interrupt moderation by the EITR counter.

Each time a Low Latency Interrupt is fired, the EITR interval is loaded and down-counting starts again.

The logic of the low latency interrupt mechanism is as follows:

- There are 8 5-tuples filters. The content of each filter is described in Section 7.1.1.5. The immediate interrupt action of each filter can be enabled or disabled. If one of the filters detects an adequate packet, an immediate interrupt is issued.
- When VLAN priority filtering is enabled, VLAN packets must trigger an immediate interrupt when the VLAN Priority is equal to or above the VLAN priority threshold. This is regardless of the status of the 5-tuple filters.
- The SYN packets filter defined in Section 7.1.1.6 and the ethernet type filters defined in section Section 7.1.1.4 might also be used to indicate low latency interrupt conditions.

Note: Immediate interrupts are available only when using advanced receive descriptors and not for legacy descriptors.

Packets that are dropped or have errors do not cause a Low Latency Interrupt.

7.3.6.1 Rate Control Mechanism

In a network with lots of latency sensitive traffics the Low Latency Interrupt can eliminate the Interrupt throttling capability by flooding the Host with too many interrupts (more than the Host can handle).

In order to mitigate the above, Intel® 82576 GbE Controller supports a credit base mechanism to control the rate of the Low Latency Interrupts.

Rules:

- The default value of each counter is 0b (no moderation). This also preserves backward compatibility.
- The counter increments at a configurable rate, and saturates at the maximum value (31d).
  - — The configurable rate granularity is 4 μs (250K interrupt/sec. down to 250K/32 ~ 8K interrupts per sec.).
- A LLI might be issued as long as the counter value is strictly positive (> zero).
  - — The credit counter allows bursts of low latency interrupts but the interrupt average are not more than the configured rate.
- Each time a Low Latency Interrupt is fired the credit counter decrements by one.
- Once the counter reaches zero, a low latency interrupt cannot be fired
  - — Must wait for the next ITR expired or for the next incrementing of this counter (if the EITR expired happened first the counter does not decrement).

The following fields manages rate control of LLI:

- The *LL Interval* field in the GPIE register controls the rate of credits.
- The 5-bit *LL Counter* field in the EITR register contains the credits

7.3.7 TCP Timer Interrupt

7.3.7.1 Introduction

In order to implement TCP timers for IOAT, software needs to take action periodically (every 10 milliseconds). Today, the driver must rely on software-based timers, whose granularity can change from platform to platform. This software timer generates a software NIC interrupt, which then allows the driver to perform timer functions as part of its usual DPC, avoiding cache thrash and enabling parallelization. The timer interval is system-specific.

It would be more accurate and more efficient for this periodic timer to be implemented in hardware. The driver would program a timeout value (usual value of 10 ms), and each time the timer expires, hardware sets a specific bit in the EICR. When an interrupt occurs (due to normal interrupt moderation schemes), software reads the EICR and discover that it needs to process timer events during that DPC.

The timeout should be programmable by the driver, and the driver should be able to disable the timer interrupt if it is not needed.

7.3.7.2 Description

A stand-alone down-counter is implemented. An interrupt is issued each time the value of the counter is zero.

The software is responsible for setting initial value for the timer in the *Duration* field. Kick-starting is done by writing a 1b to the *KickStart* bit.

Following kick-starting, an internal counter is set to the value defined by the *Duration* field. Then the counter is decreased by one each millisecond. When the counter reaches zero, an interrupt is issued (see EICR register Section 8.8.1). The counter re-start counting from its initial value if the *Loop* field is set.

7.4 802.1q VLAN Support

The 82576 provides 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.

7.4.1 802.1q VLAN Packet Format

The following table compares an untagged 802.3 Ethernet packet with an 802.1q VLAN tagged packet.

Table 7-49. Comparing Packets

802.3 Packet	#Octets		802.1q VLAN Packet	#Octets
DA	6		DA	6
SA	6		SA	6
Type/Length	2		802.1q Tag	4
Data	46-1500		Type/Length	2
CRC	4		Data	46-1500
			CRC*	4

Note: 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. Also, max frame size for an 802.1q VLAN packet is 1522 octets as opposed to 1518 octets for a normal 802.3z Ethernet packet.

7.4.2 802.1q 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 makes 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:

- User Priority (UP)
- Canonical Form Indicator (CFI). Should be 0b for transmits. For receives, the device has the capability to filter out packets that have this bit set. See the CFIEN and CFI bits in the RCTL described in Section 8.10.1.
- VLAN Identifier (VID)

Bit ordering is shown below.

Table 7-50. TCI Bit Ordering

Octet 1								Octet 2							
UP				VID											

7.4.3 Transmitting and Receiving 802.1q Packets

7.4.3.1 Adding 802.1q Tags on Transmits

Software might command the 82576 to insert an 802.1q VLAN tag on a per packet or per flow basis. If CTRL.VME is set to 1b, and the *VLE* bit in the transmit descriptor is set to 1b, then the 82576 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 VET register. 8021.Q tag insertion is done in different ways for legacy and advanced Tx descriptors:

- Legacy Transmit Descriptors:, The Tag Control Information (TCI) of the 802.1q tag comes from the *VLAN* field (see Figure 7-9) of the descriptor. Refer to Table 7-26, for more information regarding hardware insertion of tags for transmits.
- Advanced Transmit Descriptor: The Tag Control Information (TCI) of the 802.1q tag comes from the VLAN *Tag* field (see Table 7.2.2.2.1) of the advanced context descriptor. The *IDX* field of the advanced Tx descriptor should be set to the adequate context.

7.4.3.2 Stripping 802.1q Tags on Receives

Software might instruct the 82576 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), then the 82576 strips the 4 byte VLAN tag from the packet, and stores the TCI in the VLAN *Tag* field (see Figure 7-5 and See “Receive UDP Fragmentation Checksum) of the receive descriptor.

The 82576 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, but the VLAN tag is not stripped and the *VP* bit is not set. Refer Table 7-19 for more information regarding receive packet filtering.

7.4.4 802.1q 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 Ether Type (VET) register. If the VLAN type field in the incoming packet matches the VET register, the packet is then compared against the VLAN Filter Table Array for acceptance

The 82576 provides exact VLAN filtering for VLAN tags for host traffic and VLAN tags for manageability traffic.

The *Virtual LAN ID* field indexes a 4096 bit vector. If the indexed bit in the vector is one; there is a Virtual LAN match. Software might set the entire bit vector to ones if the node does not implement 802.1q filtering. The register description of the VLAN Filter Table Array is described in detail in Section 8.10.19.

In summary, the 4096-bit vector is comprised of 128, 32-bit registers. 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.

The BMC configures the 82576 with eight different manageability VIDs via the Management VLAN TAG Value [7:0] - MAVTV[7:0] registers and enables each filter in the MFVAL register.

Two other bits in the Receive Control register (see Section 8.10.1), 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 acceptance criteria for the packet.

Note: The *VFE* bit does not effect whether the VLAN tag is stripped. It only effects whether the VLAN packet passes the receive filter.

The following table lists reception actions per control bit settings.

Figure 7-19. Packet Reception Decision Table

Is packet 802.1q?	CTRL. VME	RCTL. VFE	Action
No	X	X	Normal packet reception
Yes	0b	0b	Receive a VLAN packet if it passes the standard MAC address filters (only). Leave the packet as received in the data buffer. *VP* bit in receive descriptor is cleared.
Yes	0b	1b	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 would not be stripped). *VP* bit in receive descriptor is cleared.
Yes	1b	0b	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. Sets *VP* bit in receive descriptor.
Yes	1b	1b	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. Sets *VP* bit in receive descriptor.

Note: A packet is defined as a VLAN/802.1q packet if its type field matches the VET.

7.4.5 Double VLAN Support

The 82576 supports a mode where all received and sent packet have at least one VLAN tag in addition to the regular tagging which might optionally be added. This mode is used for systems where the switches add an additional tag containing switching information.

This mode is activated by setting *CTRL_EXT.EXTENDED_VLAN* bit. The default of this bit is set according to bit 1 in word 24h/14h of the EEPROM for ports 0 and 1 respectively.

The type of the VLAN tag used for the additional VLAN is defined in the *VET.VET_EXT* field.

7.4.5.1 Transmit Behavior

It is expected that the driver includes the external VLAN header as part of the transmit data structure. The software may post the internal VLAN header as part of the transmit data structure or embedded in the transmit descriptor (see Section 7.2.2 for details). The 82576 does not relate to the external VLAN header other than the capability of “skipping” it for parsing of inner fields.

Note: The VLAN header in a packet that carries a single VLAN header is treated as the external VLAN.

The 82576 expects that any transmitted packet has at least the external VLAN added by the software. For those packets where an external VLAN is not present, any offload that relates to inner fields to the EtherType may not be provided.

7.4.5.2 Receive Behavior

When a port of the 82576 is working in this mode, the 82576 assumes that all packets received by this port have at least one VLAN, including packet received or sent on the manageability interface.

One exception to this rule are flow control PAUSE packets which are not expected to have any VLAN. Other packets may contain no VLAN, however a received packet that does not contain the first VLAN is forwarded to the host but filtering and offloads are not applied to this packet.

See the table below for the supported receive processing when the device is set to “Double VLAN” mode.

Stripping of VLAN is done on the second VLAN if it exists. All the filtering functions of the 82576 ignores the first VLAN in this mode.

The presence of a first VLAN tag is indicated it in the *RDESC.STATUS.VEXT* bit.

Queue assignment of the Rx packets is not affected by the external VLAN header. It may depend on the internal VLAN, MAC address or any upper layer content as described in Section 7.1.1.

Table 7-51. Receive Processing in Double VLAN Mode

VLAN Headers	Status.VEXT	Status.VP	Packet Parsing	Rx offload functions
External and internal	1	1	+	+
Internal Only	Not supported			
V-Ext	1	0	+	+
None[1]	0	0	+ (flow control only)	-

[1] A few examples for packets that may not carry any VLAN header may be: Flow control; LACP; LLDP; GMRP; 802.1x packets

7.5 Configurable LED Outputs

The 82576 implements 4 output drivers intended for driving external LED circuits per port. Each LAN device provides an independent set of LED outputs - these pins and their function are bound to a specific LAN device. Each of the four 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. Furthermore, the hardware-default configuration for all the LED outputs, can be specified via EEPROM fields, thereby supporting LED displays configurable to a particular OEM preference.

Each of the 4 LED's might be configured to use one of a variety of sources for output indication. The MODE bits control the LED source as described in Table 7-52.

The IVRT bits allow 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 (on for 200ms, then off for 200ms) 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 by a human eye.

Note: When LED Blink mode is enabled the appropriate LED Invert bit should be set to 0b.

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.

The dynamic LED modes (FILTER_ACTIVITY, LINK/ACTIVITY, COLLISION, ACTIVITY, PAUSED) should be used with LED Blink mode enabled.

7.5.1 MODE Encoding for LED Outputs

Table 7-52 lists the MODE encoding used to select the desired LED signal source for each LED output.

Table 7-52. Mode Encoding for LED Outputs

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 (de-asserted 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 82576 detects a 1-lane PCIe connection.
1101b	PAUSED	Asserted when the 82576’s transmitter is flow controlled.
1110b	LED_ON	Always high (Asserted)
1111b	LED_OFF	Always low (De-asserted)

7.6 Memory Error Correction and Detection

The 82576 main internal memories are protected by error correcting code or parity code. The larger memories are protected by an error correcting code (ECC) that can detect two errors and correct one error. The smaller memories are protected either with an error correcting code (ECC) that correct one error or a parity bit that can detect one error.

Correctable errors are silently corrected and are counted in the RPBECCSTS.Corr_err_cnt, TPBECCSTS.Corr_err_cnt, SWPBECCSTS.Corr_err_cnt, IPPBECCSTS.Corr_err_cnt, RDHESTS.Corr_err_cnt, TDHESTS.Corr_err_cnt, PRBESTS.Corr_err_cnt, PWBESTS.Corr_err_cnt or PMSIXESTS.Corr_err_cnt fields according to the memory in which the error was found.

Part of the uncorrectable errors are counted in the RPBECCSTS.Uncorr_err_cnt, TPBECCSTS.Uncorr_err_cnt, SWPBECCSTS.Uncorr_err_cnt, IPPBECCSTS.Uncorr_err_cnt, RDHESTS.Uncorr_err_cnt or TDHESTS.Uncorr_err_cnt fields according to the memory in which the error was found. The 82576 reacts to uncorrectable error detection according to the location in which the error was found:

- If the error was detected in a receive packet data in the main Rx packet buffer, the packet is sent to the host with the *RXE* bit set in the receive descriptor. This packet should be discarded by the host. This is considered as a non fatal error.
- If the error was detected in a transmit packet data in the main Tx packet buffer, the packet is sent to the network with a wrong FCS so that the link partner can discard it. This is also, considered as a non fatal error.
- If the error was detected in the descriptors attached to receive or transmit packets in the descriptor handler cache memory, or a parity error was detected in one of the internal control memories the

consistency of the receive/transmit flow can not be guaranteed any more. In this case the traffic is stopped and an interrupt is raised and the memory in which the error was detected is indicated in the PEIND register. The flow stop can be released only by software reset (CTRL.RST). This is considered as a fatal error.

- If an error is detected in the IPsec Rx SA table, the traffic is stopped and an interrupt is raised and the memory in which the error was detected is indicated in the PEIND register. The flow stop can be released only by software reset (CTRL.RST). This is considered as a fatal error.
- The interrupt causes used to indicate an error are ICR[23:22] according to the severity of the error.

Note: Once an interrupt indicating a memory error was asserted, the PEIND register must be read before a new interrupt can be asserted.

The enabling of the reaction mechanism of the 82576 to uncorrectable errors for each of the memories is done using the PEINDM register. Enablement of parity error detection is done using the PEINDM. *Parity_en* field. Enablement of ECC error correction for each memory is done using the *ECC Enable* field in the RPBECCSTS, TPBECCSTS, SWPBECCSTS, IPPBECCSTS, RDHESTS, TDHESTS, PRBESTS, PWBESTS or PMSIXESTS registers.

7.7 DCA

7.7.1 Description

Direct Cache Access (DCA) is a method to improve network I/O performance by placing some posted inbound writes directly within CPU cache. Through research and experiments, DCA has been shown to reduce CPU Cache miss rates significantly.

DCA provides a mechanism where the posted write data from an I/O device, such as an Ethernet NIC, can be placed into CPU cache with a hardware pre-fetch. This mechanism is initialized upon a power good reset. A device driver for the I/O device configures the I/O device for DCA and sets up the appropriate CPU ID and bus ID for the device to send data. The device will then encapsulate that information in PCIe TLP headers, in the *TAG* field, to trigger a hardware pre-fetch by the MCH /IOH to the CPU cache.

DCA implementation is controlled by separated registers (RXCTL and TXCTL) for each receive and transmit queues. In addition, a *DCA Enable* bit can be found in the DCA_CTRL register, and a DCA_ID register can be found for each port, in order to make visible the function, device, and bus numbers to the driver.

The RXCTL and TXCTL registers can be written by software on the fly and can be changed at any time. When software changes the register contents, hardware applies changes only after all the previous packets in progress for DCA has been completed.

However, in order to implement DCA, the 82576 has to be aware of the Crystal Beach version used. The software driver must initialize the 82576 to let be aware of the crystal Beach version. A new register named DCA_CTRL is used in order to properly define the system configuration.

There are 2 modes for DCA implementation:

1. Legacy DCA: The DCA target ID is derived from CPU ID (similar to Goshen)
2. DCA 1.0: The DCA target ID is derived from APIC ID.

The software driver selects one of these modes through the DCA_mode register.

The details of both modes are described below.

7.7.2 Details of Implementation

7.7.2.1 PCIe Message Format for DCA

Figure 7-20 shows the format of the PCIe message for DCA.

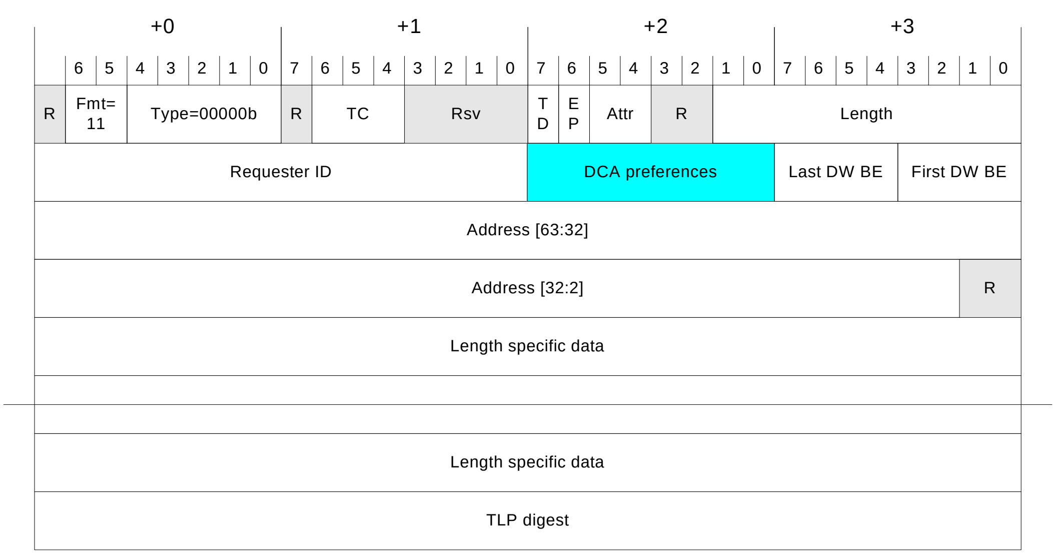


Figure 7-20. PCIe Message Format for DCA

The DCA preferences field has the following formats.

Table 7-53. Legacy DCA Systems

Bits	Name	Description
0	DCA indication	0b: DCA disabled 1b: DCA enabled
4:1	DCA Target ID	The DCA Target ID specifies the target cache for the data.
7:5	Reserved	Reserved

Table 7-54. DCA 1.0 Systems

Bits	Name	Description
7:0	DCA target ID	0000.0000b: DCA is disabled Other: Target Core Id derived from APIC Id. The method for this is described in DCA Platform Architecture Specification, section 7.3.1 (Anacapa reference number 16802)

Note: All functions within a the 82576 have to adhere to the “tag encoding” rules for DCA writes. Even if a given function is not capable of DCA, but other functions are capable of DCA, memory writes from the non-DCA function must set the *Tag* field to “00000000”.

7.8 Next Generation VMDqSecurity

7.8.1 IPSec Support

7.8.1.1 Overview

This chapter defines the hardware requirements for the IPsec off-load ability included in the 82576. IPsec off-load is the ability to handle in hardware a certain amount of the total number of IPsec flows, while the remaining are still handled by the operating system. It is the operating system responsibility to submit to hardware the most loaded flows, in order to take maximum benefits of the IPsec off-load in term of CPU utilization savings. The establishment of the IPsec Security Associations between peers is outside the scope of this document, since it always is handled by the operating system. In general, the requirements on the driver or on the operating system for enabling IPsec off-load are not detailed here.

When an IPsec flow is handled in software, since the packet might be encrypted and the integrity check field already valid (IPv4 options might be present in the packet together with IPsec headers) the 82576 processes it like it does for any other unsupported Layer4 protocol, and without performing on it any Layer4 offload.

7.8.1.1.1 Related RFCs and Other References

- RFC4106 - The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
- RFC4302 - IP Authentication Header (AH)
- RFC4303 - IP Encapsulating Security Payload (ESP)
- RFC4543 - The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
- GCM spec - McGrew, D. and J. Viega, “The Galois/Counter Mode of Operation (GCM)”, Submission to NIST. http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec.pdf, January 2004.

7.8.1.2 Hardware Features List

7.8.1.2.1 Main Features

- off-load IPsec for up to 256 Security Associations (SA) for each side separately, Tx and Rx.
  - On-chip storage for both Tx and Rx SA tables
  - Tx SA index is conveyed to hardware via Tx context descriptor
  - Rx SA lookup is a deterministic search according to a search key made of SPI, destination IP address, and IP version type (IPv6 or IPv4)
  - Performance in RX: update the whole Rx SA table in less than 1msec, while receiving back-to-back 64-bytes packets
- IPsec protocols:
  - IP Authentication Header (AH) protocol for authentication
  - IP Encapsulating Security Payload (ESP) for authentication only
  - IP Encapsulating Security Payload (ESP) for both authentication and encryption, only if using the same key for both
- Crypto engines:
  - For AH or ESP authentication only use AES-128-GMAC (128-bit key)
  - For ESP encryption and authentication use AES-128-GCM (128-bit key)
- IPsec encapsulation mode: Transport mode

  - In Tx, packets are provided by software already encapsulated with a valid IPsec header (for AH with blank ICV inside), and
    - for ESP single send, with a valid ESP trailer and ESP ICV (blank ICV)
    - for ESP large send, without ESP trailer and without ESP ICV
  - In Rx, packets are provided to software encapsulated with their IPsec header and for ESP with the ESP trailer and ESP ICV,
    - where up to 255-bytes of incoming ESP padding is supported, for peers that prefer hiding the packet length
- IP versions:
  - IPv4 packets that do not include any IP option
  - IPv6 packets that do not include any extension header (other than AH/ESP extension header)
- Rx statuses reported to software via Rx descriptor:
  - Packet type: AH/ESP
  - IPsec off-load done (SA match)
- One Rx error reported to software via Rx descriptor in the following precedence order: No error, Invalid IPsec protocol, Packet length error, Authentication failed

7.8.1.2.2 Cross Features

- w/ segmentation: full coexistence (TCP/UDP packets only)
  - increment IPsec Sequence Number (SN) and Initialization Vector (IV) on each additional segment
- w/ checksum off-load: full coexistence (Tx and Rx)
  - IP header checksum
  - TCP/UDP checksum
- w/ IP fragment: no IPsec offload done on IP fragments
- w/ RSS: full coexistence, hash on the same fields used without IPsec (either 4-tuples or 2-tuples)
- w/ virtualization:
  - Full coexistence in Next Generation VMDq mode
  - in IOV mode, all IPsec registers are owned by the VMM/PF. IPsec can be used for VMotion traffic (for instance).
  - No coexistence with VM to VM switch, IPsec packets handled in HW are not looped back by the 82576 to another VM. Tx IPsec packets destined to a local VM must be handled in SW and looped back via the SW switch. Anti-spoofing check is however performed on any IPsec packet.
- w/ 9.5KB Jumbo frames: full coexistence
- w/ 802.1x: no interaction
- w/ teaming: no interaction
- w/ TimeSync:
  - TimeSync 1588v1 UDP packets must not be encapsulated as IPsec packets
  - No interaction with TimeSync 1588v2 Layer2 packets
- w/ Layer2 encapsulation modes:
  - IPsec offload is not supported for flows with SNAP header
  - IPsec offload will coexist with double-VLAN encapsulations
- Tunneled IPsec packets in receive: IPsec offload supported, but no other Layer4 offload performed
- w/ NFS: NFS packets encapsulated over ESP packets (whether offloaded or not) are not recognized

- w/ SCTP offload: no SCTP CRC32 off load performed on received ESP packets (even those handled by HW), but SCTP offload performed on any IPsec packet.
- w/ Manageability traffic: IPsec offload ability is controlled exclusively by the host, and because of an implementation limitation, no IPsec offload is possible on Tx MNG packets. For IPsec flows handled by software,
  - — If manageability and host entities share some IP address(es), then manageability should coordinate any use of IPsec protocol with the host. Note it should be true for previous devices that do not offer IPsec offload.
  - — If manageability and host entities have totally separate IP addresses, then manageability can use IPsec protocol (as long as it is handled by the BMC software).
- w/ header split/replication:
  - — For SAs handled in hardware, IP boundary split is done before the IPsec header
  - — For SAs handled in software, no header split/replication done
- w/ 5-tuple Rx filters:
  - — ESP packets recognizes only TCP, UDP, and SCTP protocols in FTQF registers

7.8.1.3 Software/Hardware Demarcation

The followings items are not supported by the hardware but might be supported by operating system/Driver:

- Multicast SAs
- IPsec protocols:
  - — Both AH and ESP protocols on the same SA or packet
  - — ESP for encryption only
  - — ESP for both authentication and encryption using different keys and/or different crypto engines
- Crypto engines:
  - — AES-256, SHA-1, AES-128-CBC, or any other crypto algorithm
- Tx IPsec packets encapsulated in Tunnel mode
- Extended Sequence Number (ESN)
- IP versions:
  - — IPv4 packets that include IP option
  - — IPv6 packets that include extension headers other than the AH/ESP extension headers
- Anti-replay check and discard of incoming replayed packets
- Discard of incoming “dummy” ESP packets (packets with protocol value 59)
- IPsec packets that are IP fragments
- ESP padding content check
- IPsec statistics
- IPsec for flows with SNAP header

Note: For SCTP and other Layer4 header types, or for tunneled packets, HW should not care what is there when doing Rx IPsec processing. Everything after the IP/IPsec headers may be opaque to HW - just think of it as IP payload. It is fine to do IPsec processing on any packet that has a matching SA and appropriate IP options/extension headers.

There is no expectation that HW figure out what is in the packet beyond the IP/IPsec headers before decrypting/authenticating the packet. The most important point is that HW should not corrupt or drop incoming IPsec packets - in any situation. When HW decides and start performing IPsec offload on a packet, it should pursue the offload until packet's end - at the price of eventually not doing other Layer3/4 off loads on it. It is always legitimate for HW not to start doing the IPsec offload on a matched SA, if it knows it is an unsupported encapsulation - i.e. one of the 3 cases: IPv4 option, IPv6 extensions, or SNAP.

7.8.1.4 IPsec Formats Exchanged Between Hardware and Software

This section deals with the IPsec packet encapsulation formats used between software and hardware by IPsec packet concerned with the off-load in either Tx or Rx direction.

In Rx direction, the IPsec packets are delivered by hardware to software encapsulated as they were received from the line, whether IPsec off-load was done or not, and when it was done, whether authentication/decrypting has succeeded or failed.

7.8.1.4.1 Single Send

In Tx direction, single send IPsec packets are delivered by software to hardware already encapsulated and formatted with their valid IPsec header and trailer contents, as they should be run over the wire - excepted to the *ICV* field that is filled with zeros, and to the ESP payload destined to be encrypted that is provided in clear text before any encryption.

7.8.1.4.2 Single Send With TCP/UDP Checksum Offload

For single send ESP packets with TCP/UDP checksum off-load, the checksum computing must include the TCP/UDP header and payload before hardware encryption occurred and without the ESP trailer and ESP ICV provided by software. Software provides the length of the ESP trailer plus ESP ICV in a dedicated field of the Tx context descriptor (*IPS_ESP_LEN* field) to allow hardware know when to stop TCP/UDP checksum computing.

Software calculates a full checksum for the IP pseudo-header as in the usual case. The protocol value used in the IP pseudo-header must be the TCP/UDP protocol value and not the AH/ESP protocol value that appears in the IP header. This full checksum of the pseudo-header is placed in the packet data buffer at the appropriate offset for the checksum calculation.

The byte offset from the start of the DMA'd data to the first byte to be included in the TCP/UDP checksum. For example, the start of the TCP header, is computed as in the usual case: MACLEN+IPLEN. It assumes that IPLEN provided by software in the Tx context descriptor is the sum of the IP header length with the IPsec header length.

Note: For the IPv4 header checksum off-load, hardware could no more rely on the *IPLEN* field provided by software in the Tx context descriptor, but should rely on the fact that no IPv4 options is present in the packet. Consequently, for IPsec off-load packets hardware computes the IP header checksum over always a fixed amount of 20-bytes.

7.8.1.4.3 Large Send TCP/UDP

In Tx direction, large send IPsec packets are delivered by software to hardware already encapsulated and formatted with only their valid IPsec header contents - excepted to the *ICV* field included in AH packets headers that is filled with zeros, and to the ESP payload destined to be encrypted that is provided in clear text before any encryption. No ESP trailer or ESP ICV are appended to the large send by software. It means that hardware has to append the ESP trailer and ESP ICV on each segment by itself, and to update IP total length / IP payload length accordingly.

The next header of the ESP trailer to be appended by hardware is taken from TUCMD.L4T field of the Tx context descriptor.

By definition large send segmentation requires on each segment that the IP total length / IP payload length be updated, and the IP header checksum and TCP/UDP checksum be re-computed. But for the large send of IPsec packets, the SN and the IV fields must be increased by one in hardware on each new segment (after the first one) as well.

Caution: Driver should not offload a large send that will cause SN and/or IV field to wrap-around in HW.

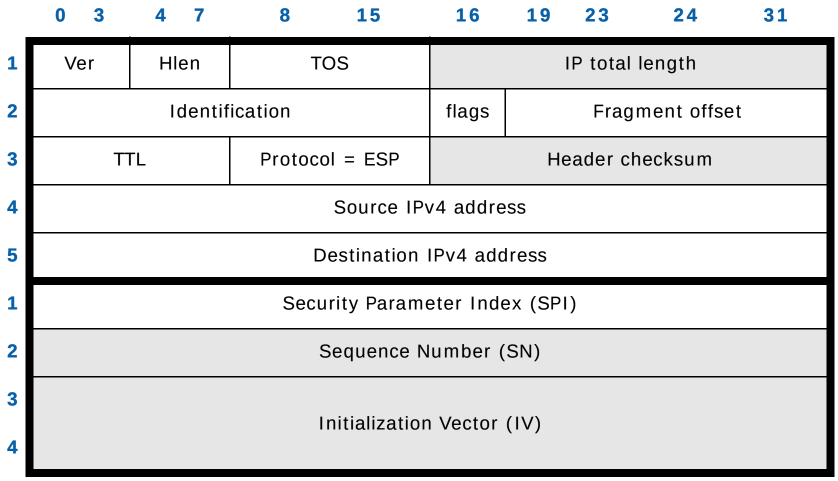
Software calculates a partial checksum for the IP pseudo-header as in the usual case. The protocol value used in the IP pseudo-header must be the TCP/UDP protocol value and not the AH/ESP protocol value that appears in the IP header. This partial checksum of the pseudo-header is placed in the packet data buffer at the appropriate offset for the checksum calculation.

The byte offset from the start of the DMA'd data to the first byte to be included in the TCP/UDP checksum. For example, the start of the TCP/UDP header, is computed as in the usual case: MACLEN+IPLEN. It assumes that IPLEN provided by software in the Tx context descriptor is the sum of the IP header length with the IPsec header length.

For large send ESP packets, the TCP/UDP checksum computing must include the TCP/UDP header and payload before hardware encryption occurred and without the ESP trailer and ESP ICV appended by hardware. Hardware must stop TCP/UDP checksum computing after the amount of bytes given by L4LEN + MSS. It is assumed that MSS value placed by software in the Tx context descriptor specifies the maximum TCP/UDP payload segment sent per frame, not including any IPsec header or trailer - and not including the TCP/UDP header.

Note: For IPv4 header checksum computing, refer to the note in section Section 7.8.1.4.2.

Shaded fields in the figures below correspond to fields that need to be updated per each segment.



	0 3	4 7	8 15	16	19 23	24 31
1	Ver	Hlen	TOS	IP total length		
2	Identification			flags	Fragment offset	
3	TTL		Protocol = ESP	Header checksum		
4	Source IPv4 address					
5	Destination IPv4 address					
1	Security Parameter Index (SPI)					
2	Sequence Number (SN)					
3	Initialization Vector (IV)					
4						

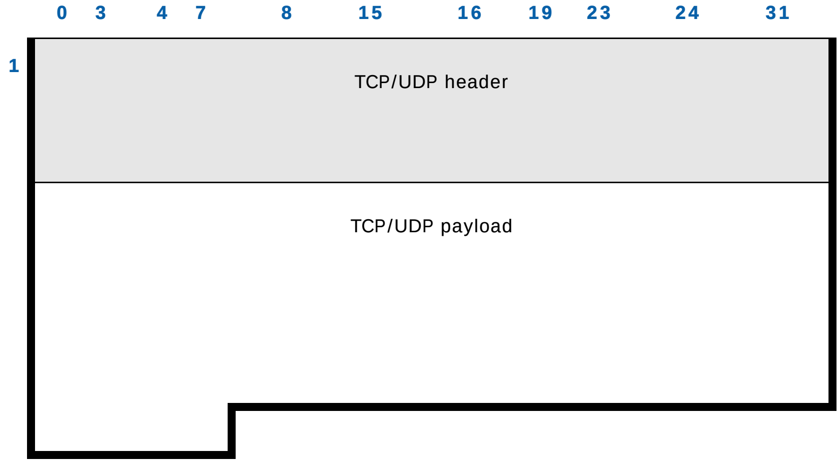
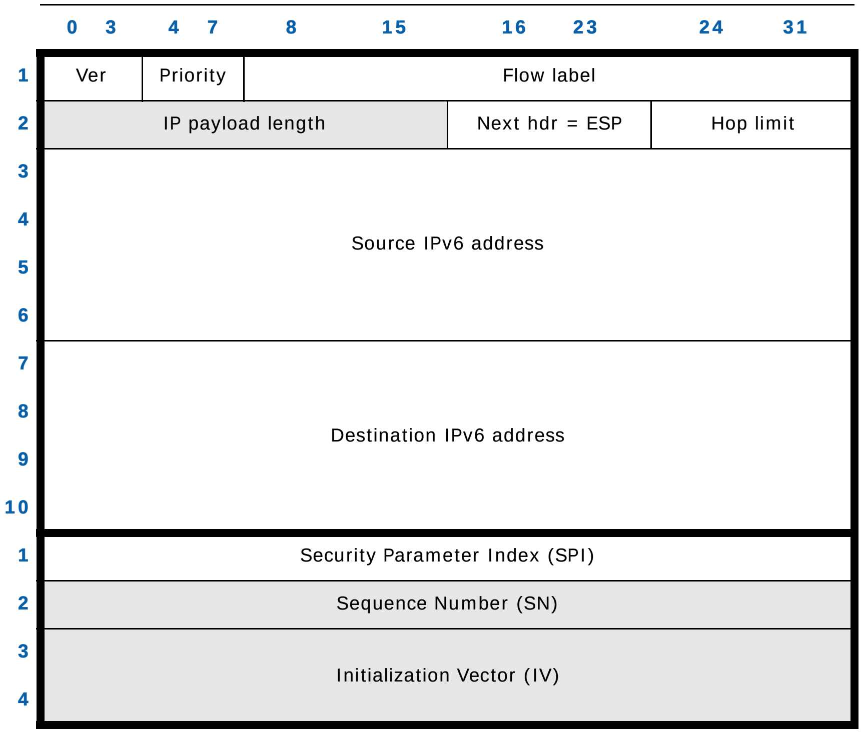


Figure 7-21. IPv4 Large Send ESP Packet Provided by Software



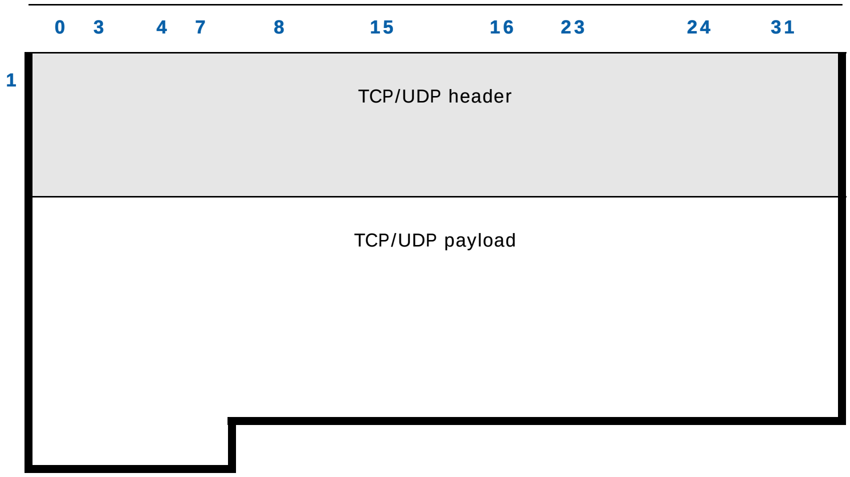


Figure 7-22. IPv6 Large Send ESP Packet Provided by Software

7.8.1.5 TX SA Table

IPsec off-load is supported only via advanced transmit descriptors. See Section 7.2.2 for details.

7.8.1.5.1 Tx SA Table Structure

The Tx SA table contains the extra info required by the AES-128 crypto engine to authenticate and encrypt the data. This info is not run over the wire together with the IPsec packets, but it is rather exchanged between the IPsec peers' operating system during the Security Association establishment process. When the IKE software does a key computation it computes 4 extra bytes using a pseudo-random function, i.e it generates 20 bytes instead of 16 bytes that it needs to use as a key – and the last 4 bytes are used as salt value.

The SA table in Tx is a 256 x 20-bytes table loaded by software, each line in the table contains the following fields:

Table 7-55. TX SA Table

AES-128 KEY	AES-128 SALT
16 bytes	4 bytes

Refer to Section 7.8.1.7 for a description of the way these fields are used by the AES-128 crypto engine.

Whenever an unrecoverable memory error occurs when accessing the Tx SA tables, an NFER interrupt is generated in the ICR register, the index of the corrupted SA entry is reported via the PEIND register, and the transmit path is stopped until the host resets the device. Packets that have already started to be transmitted on the wire is sent with a wrong CRC.

Upon reset, the 82576 clears the contents of the Tx SA table.

7.8.1.5.2 Access to Tx SA Table

1. Software writes the IPSTXIDX register.
2. Software writes the IPSTXKEY 0..3 first, and then write IPSTXSALT at the end, as it is used internally to trigger the write of the whole entry into the TX SA Table. Read access to these registers can be done in any order.

3. Hardware issues a write/read command to/from the SA table copying/reading the IPSTXKEY (16 bytes) and the IPSTXSALT (4 bytes) to/from the SA table using the index in the IPSTXIDX.
4. Software starts/resumes sending IPsec off-load packets with SA_IDX pointing to valid/invalid SA entries. A valid SA entry contains updated key and salt fields currently in use by the IPsec peers.

7.8.1.6 TX Hardware Flow

7.8.1.6.1 Single Send Without TCP/UDP Checksum Offload:

1. Extract IPsec off-load request from the *IPSEC* bit of the *POPTS* field in the advanced Tx transmit data descriptor.
2. If IPsec off-load is required for the packet (*IPSEC* bit was set), then extract the SA_IDX, ENCRYPTION, and IPSEC_TYPE fields from the Tx context descriptor associated to that flow.
3. Fetch the AES-128 KEY and SALT from the Tx SA entry indexed by SA_IDX, and according to the ENCRYPTION and IPSEC_TYPE bits determine which IPsec off-load to perform.
4. For AH, zero the mutable fields
5. Compute ICV and encrypt data (if required for ESP) over the appropriate fields according to the operating rules described in Section 7.8.1.7, and making use of the AES-128 KEY and SALT fields fetched at step 3.
6. Insert ICV at its appropriate location and replace the plaintext with the ciphertext (if required for ESP).

7.8.1.6.2 Single Send With TCP/UDP Checksum Offload:

1. Extract IPsec off-load command from the *IPSEC* bit of the *POPTS* field in the advanced Tx transmit data descriptor.
2. If IPsec off-load is required for the packet (*IPSEC* bit was set), then extract the SA_IDX, ENCRYPTION, IPSEC_TYPE, and IPS_ESP_LEN fields from the Tx context descriptor associated to that flow.
3. Fetch the AES-128 KEY and SALT from the Tx SA entry indexed by SA_IDX, and according to the ENCRYPTION and IPSEC_TYPE bits determine which IPsec off-load to perform.
4. Compute the byte offset from the start of the DMA'd data to the first byte to be included in the checksum (the start of the TCP header, as specified in Section 7.8.1.4.2).
5. Compute TCP/UDP checksum until either the last byte of the DMA data or for ESP packets, up to IPS_ESP_LEN bytes before it. Like for the usual case, *implicitly* pad out the data by one zeroed byte if its length is an odd number.
6. Sum the full checksum of the IP pseudo header placed by software at its appropriate location with the TCP/UDP checksum computed at step 5. Overwrite the checksum location with the one's complement of the sum.
7. For AH, zero the mutable fields
8. Compute ICV and encrypt data (if required for ESP) over the appropriate fields according to the operating rules described in Section 7.8.1.7, and making use of the AES-128 KEY and SALT fields fetched at step 3.
9. Insert ICV at its appropriate location and replace the plaintext with the ciphertext (if required for ESP).

7.8.1.6.3 Large Send TCP/UDP:

1. Extract IPsec off-load command from the *IPSEC* bit of the *POPTS* field in the advanced Tx transmit data descriptor.
2. If IPsec off-load is required for the packet (*IPSEC* bit was set), then extract the SA_IDX, ENCRYPTION, and IPSEC_TYPE fields from the Tx context descriptor associated to that flow.

3. Fetch the AES-128 KEY and SALT from the Tx SA entry indexed by SA_IDX, and according to the ENCRYPTION and IPSEC_TYPE bits determine which IPsec off-load to perform.
4. Fetch the packet header from system memory, up to IPLEN+L4LEN bytes from the start of the DMA'd data.
5. Overwrite the TCP sequence number with the stored accumulated TCP sequence number (if it is not the first segment).
6. Fetch (next) MSS bytes (or the remaining bytes up to PAYLEN for the last segment) from the system memory and form the segment formed by packet header and data bytes, while storing the accumulated TCP sequence number.
7. Compute the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, i.e the start of the TCP/UDP header, as specified in Section 7.8.1.4.3.
8. Compute TCP/UDP checksum until the last byte of the DMA data. Like for the usual case, *implicitly* pad out the data by one zeroed byte if its length is an odd number.
9. For both IPv4 and IPv6, hardware needs to factor in the TCP/UDP length (typically L4LEN+MSS) to the software supplied pseudo header partial checksum. Then it sums the so obtained full checksum of the IP pseudo header with the TCP/UDP checksum computed at step 7. Overwrite the TCP/UDP checksum location with the one’s complement of the sum.
10. Increment by one the AH/ESP SN and IV fields on every segment (excepted to the first segment), and store the updated SN and IV fields with other temporary statuses stored for that large send (one large send set of statuses per Tx queue).
11. For ESP, append the ESP trailer: 0-3 Padding bytes, Padding length, and Next header = TCP/UDP protocol value, in a way to get the 4-bytes alignment as described in Section 7.8.1.4.3.
12. Compute the IP total length / IP payload length and compute IPv4 header checksum as described in the note of section Section 7.8.1.4.1. Place the results in their appropriate location.
13. For AH, zero the mutable fields
14. Compute ICV and encrypt data (if required for ESP) over the appropriate fields according to the operating rules described in Section 7.8.1.7, and making use of the AES-128 KEY and *SALT* field fetched at step 3.
15. Insert ICV at its appropriate location and replace the plaintext with the ciphertext (if required for ESP).
16. Go back to step 4 to process the next segment (if necessary).

7.8.1.7 AES-128 Operation in Tx

The AES-128-GCM crypto engine is used for IPsec. It is referred throughout the document as an AES-128 black box, with 4-bit string inputs and 2-bit string outputs, as shown in the figure below. Refer to the GCM spec for the internal details of the engine. The difference between different IPsec modes resides in the set of inputs presented to the box.

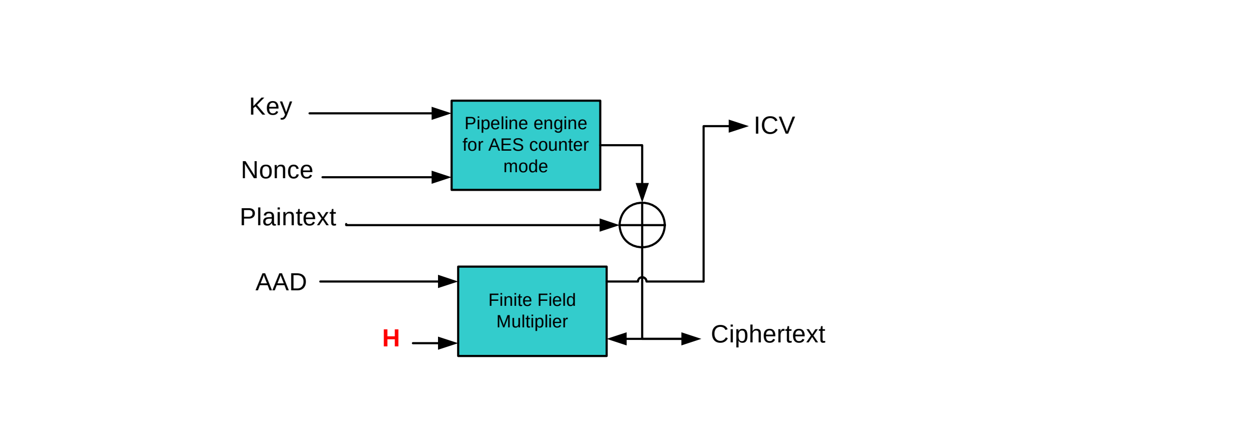


Figure 7-23. AES-128 Crypto Engine Box

- **Key** - 128-bits AES-128 KEY field (secret key) stored for that IPsec flow in the Tx SA Table:

  Key = AES-128 KEY

- **Nonce** - 96-bits initialization vector used by the AES-128 engine, which must be distinct for each invocation of the encryption operation for a fixed key. It is formed by the *AES-128 SALT* field stored for that IPsec flow in the Tx SA Table, appended with the *Initialization Vector (IV)* field included in the IPsec packet:

  Nonce = [AES-128 SALT, IV]

  The Nonce, also confusingly referred as IV in the GCM spec, is broken into two pieces - a fixed random part “salt” and increasing counter part IV, so the salt value goes with the packet as the fixed part. The purpose behind using the ‘salt’ value is to prevent offline dictionary-type attacks in hashing case, to prevent predictable patterns in the hash.
- **AAD** - Additional Authentication Data input, which is authenticated data that must be left un-encrypted.
- Plaintext - Data to be both authenticated & encrypted.
- **Ciphertext** - Encrypted data, whose length is exactly that of the plaintext.
- **ICV** - 128-bits Integrity Check Value (referred also as authentication tag).
- **H** - is internally derived from the key.

Note: The square brackets in the formulas is used as a notation for concatenated fields.

7.8.1.7.1 AES-128-GCM for ESP - Both Authenticate and Encrypt

AAD = [SPI, SN]
Plaintext = [TCP/UDP header, TCP/UDP payload, ESP trailer]

Note: Unlike other IPsec modes, in this mode, *IV* field is used only in the Nonce, and it is not included in either the Plaintext or the AAD inputs.

ESP trailer does not include the *ICV* field.

7.8.1.7.2 AES-128-GMAC for ESP - Authenticate Only

AAD = [SPI, SN, IV, TCP/UDP header, TCP/UDP payload, ESP trailer]
Plaintext = [] = empty string, no plaintext input in this mode

Note: ESP trailer does not include the *ICV* field.

7.8.1.7.3 AES-128-GMAC for AH - Authenticate Only

AAD = [IP header, AH header, TCP/UDP header, TCP/UDP payload]
Plaintext = []= empty string, no plaintext input in this mode

Note: Both IP header and AH header contain mutable fields that must be zeroed prior to be entered into the engine. Among other fields, the AH header always includes SPI, SN, and IV fields.

7.8.1.8 RX Descriptors

IPsec off-load is supported only via advanced receive descriptors. See Section 7.1.5 for details.

7.8.1.9 Rx SA Table

7.8.1.9.1 Rx SA Table Structure

The Rx SA table contains the extra info required by the AES-128 crypto engine to authenticate and decrypt the data. This info is not run over the wire together with the IPsec packets, but it is rather exchanged between the IPsec peers' operating system during the Security Association establishment process. When the IKE software does a key computation it computes 4 extra bytes using a pseudo-random function, i.e it generates 20 bytes instead of 16 bytes that it needs to use as a key – and the last 4 bytes are used as salt value.

The SPI is allocated by the receiving operating system in a unique manner. However, in a virtualized context, guest operating systems can allocate SPI values that collide with the SPI values allocated by the VMM/PF. Consequently, for enabling IPsec off-load at least for the VMM/PF flows in such situations, the SPI search must be completed by comparing the destination IP address with the IP address of the VMM/PF. Also, handling a separate table for storing the IP addresses of the VMM/PF would cost a similar die-size than storing an IP address per each SA entry, which is the selected architecture. By doing so, guest operating system could now use the proposed IPsec off-load in Rx, suffice it their SAs are configured via the VMM/PF. It is assumed that refreshing the SAs would be done once every several minutes, and would thus not overload the VMM/PF.

The SA table in Rx is a 256 x 40-bytes table loaded by software but lines are reordered by hardware. Each line in the table looks as follow:

Table 7-56. RX SA Table

SPI	IP address	IPsec Mode	AES-128 KEY	AES-128 SALT
4 bytes	16 bytes	1 byte	16 bytes	4 bytes

IPsec Mode field contains the following bits taken from IPSRXCMD register:

— IPv6, DECRYPT, and PROTO

To allow performing on it a binary search, the table above is sorted by hardware with the use of the concatenated [SPI, IP address, IPv6 bit] sort/search key:

- After reset or after a DELETE_ALL command, the hardware deletes all the entries from the Rx table.
- The software can add/delete an entry (one at a time) – using a lock mechanism.
- The hardware keeps the table sorted upon any entry addition/deletion.

- When the hardware is in a middle of deleting or adding an entry, since it uses a temporary copy of the entry it can still process incoming traffic and can do the lock up in a higher priority.
- Hardware tracks the number of used SA for debug purpose and for eventually limiting the look up search only into the used SA (for example if only just 11 SA are used, gurney to have them at the beginning of the table, the hardware starts the look up in the first 16 entries only – that takes 4 cycles instead of 8).

In the normal operating mode, the table is handled by the hardware, and software can only add/delete entries - without directly accessing the table contents. For debugging purposes, a direct read access to the table is provided. The debugging mode is enabled by setting the *DBG_MOD* bit in the IPSRXIDX register, and this access mode has precedence over the normal access mode.

Whenever an unrecoverable memory error occurs when accessing the Rx SA table, an NFER interrupt is generated in the ICR register, the index of the corrupted SA entry is reported via the PEIND register, and the receive path is stopped until the host resets the device.

Upon reset, the 82576 clears the contents of the Rx SA table.

7.8.1.9.2 Normal Access to Rx SA Table

1. Software polls the *IPSRXCMD.BUSY* bit (read access).
2. If the bit is cleared software writes to the IPSRXKEY 0..3, IPSRXSALT, IPSRXSPI, and IPSRXIPADDR 0..3 registers with the relevant SA fields to be added or deleted.
3. Software writes to the IPSRXCMD register. Setting the SA mode and the right value in the *ADD_DEL* bit. This write triggers the hardware to start insertion/deletion of the entry into the SA table.
4. Hardware searches the appropriate location of the SA entry based on the SPI, IP address and IP version bit (IPv6/IPv4).
5. Hardware issues a add/delete process of this entry and set the *Busy* bit.
   a. Add – hardware pushes all the entries below the found location, and inserts the new one, also increments the *USED_SA* field.
   b. Delete – hardware pops all the entries below the founded entry and override it, also decrements the *USED_SA* field.
   c. During the add/delete process hardware stores the intermediate SA. When incoming packet is processed, the add/delete process halts, and a different lookup process takes place using the current packet parameters (SPI, IP address and IP version). This look up starts just if there is no match with the intermediate SA.
6. Hardware clears the *Busy* bit – software can issue a new access.

7.8.1.9.3 Debugging Read Access to Rx SA Table

1. Software polls the *IPSRXCMD.BUSY* bit (read access).
2. If the bit is cleared, software writes the IPSRXIDX register with the index in the Rx SA to be read, while enabling the debugging access mode via setting the DBG_MOD bit.
3. Software read the IPSRXKEY 0..3, IPSRXSALT, IPSRXSPI, IPSRXIPADDR 0..3, and IPSRXCMD registers.
4. Hardware issues a read command to/from the SA table reading the registers from the SA entry indexed by the IPSRXIDX content.
5. Software disables the debugging access mode (by clearing the DBG_MOD) or performs another read access from step 1 above. Note that incoming packets can still be processed while in the debugging read access mode.

7.8.1.10 RX Hardware Flow Without TCP/UDP Checksum Offload

1. Detect the presence of an IPsec header and determine its type AH/ESP.

2. If an IPsec header is present (as announced by the *IP Protocol* field for IPv4 or by the next header for IPv6), then extract the SPI, destination IP address, and IP protocol (IPv4 or IPv6), and use these fields as the search key in the Rx SA table. Also report the IPsec protocol found in the Security bits of the *Extended Status* field in the advanced Rx descriptor.
3. If an SA entry matches with the search key, then fetch the IPsec Rx mode from the SA entry, and according to the PROTO and DECRYPT bits determine which IPsec off-load to perform. Also, set the *SECP* bit of the *Extended Status* field in the advanced Rx descriptor. If there was no SA match, then clear the *SECP* bit, report no error in Security error bits of the *Extended Errors* field in the advanced Rx descriptor, and stop processing the packet for IPsec.
4. If the *PROTO* field recorded in the Rx SA table does not match the *IP Protocol* field (next header for IPv6) seen in the packet, then report it via the Security error bits of the *Extended Errors* field in the advanced Rx descriptor, and stop processing the packet for IPsec.
5. Fetch the AES-128 KEY and SALT from the matched Rx SA entry.
6. For AH, zero the mutable fields
7. Check AH/ESP header is not truncated, and for ESP, check whether the packet is 4-bytes aligned. If it is not report it via the Security error bits of the *Extended Errors* field in the advanced Rx descriptor, but processing of the packet for IPsec may be completed (if it has already started). A truncated IPsec packet is a valid Ethernet frame (at least 64B) shorter than:
   a. ESP – at least 40 bytes following the IP header (16 [ESP header] + 4 [min. padding, pad_len, NH] + 16 [ICV] + 4 [CRC])
   b. AH over IPv4 – at least 40 bytes following the IP header (20 [AH header] + 16 [ICV] + 4 [CRC])
   c. AH over IPv6 – at least 44 bytes following the IP header (20 [AH header] + 4 [ICV padding] + 16 [ICV] + 4 [CRC])
8. Compute ICV and decrypt data (if required for ESP) over the appropriate fields according to the operating rules described in Section 7.8.1.12, and making use of the *AES-128 KEY* and *SALT* field fetched at step 5.
9. Compare the computed ICV with the *ICV* field included in the packet at its appropriate location and report the comparison status match/fail via the Security error bits of the *Extended Errors* field in the advanced Rx descriptor.

7.8.1.11 RX Hardware Flow With TCP/UDP Checksum Offload

Perform the RX hardware flow described in Section 7.8.1.10 above and add the following steps:

10. Start computing the checksum from the TCP/UDP header’s beginning - found according to the Rx parser logic updated for IPsec formats.
11. For ESP, stop checksum computing before the beginning of the ESP trailer - found from the end of packet according to the *Padding Length* field content, and to the formats. Like for the usual case, *implicitly* pad out the data by one zeroed byte if its length is an odd number.
12. Store the next header extracted from the AH header/ESP trailer into the *Packet Type* field of the advanced Rx descriptor, but use the TCP/UDP protocol value in the IP pseudo header used for the TCP/UDP checksum. Also compute the TCP/UDP packet length to be inserted in the IP pseudo header (excluding any IPsec header or trailer).
13. Compare the computed checksum value with the TCP/UDP checksum included in the packet. Report the comparison status in the *Extended Errors* field of the advanced Rx descriptor.

7.8.1.12 AES-128 Operation in Rx

The AES-128 operation in Rx is similar to the operation in Tx, while for decrypting the encrypted payload is fed into the plaintext input, and the resulted ciphertext stands for the decrypted payload. Refer to Section 7.8.1.7 for the proper inputs to use in every IPsec mode.

7.8.1.13 Handling IPsec Packets in Rx

The following table summarizes how IPsec packets are handled according to some of their characteristics.

Table 7-57. Summary of IPsec Packets Handling in Rx

IP fragment	IPv4 option or IPv6 extensions or SNAP	IP version	SA match	IPsec offload in HW	Layer4/3 offload in HW	Header split	AH/ESP reported in Rx desc.
yes	yes	v4	don't care	no	IP checksum only	up to IPsec header **excluded**	yes
yes	yes	v6	don't care	no	no	up to IP fragment extension included	no
yes	no	v4	don't care	no	IP checksum only	up to IPsec header **excluded**	yes
no	yes	v4	don't care	no	IP checksum only	no	yes
no	yes	v6	don't care	no	no	up to first unknown or IPsec extension header, **excluded**	no[1]
no	no	v4	yes	yes	yes[2]	up to TCP/UDP/SCTP[3] header included, no split otherwise	yes
no	no	v4	no	no	IP checksum only	no	yes
no	no	v6	yes	yes	yes[4]	up to TCP/UDP/SCTP[5] header included, no split otherwise	yes
no	no	v6	no	no	no	no	yes

[1] Exception to SNAP IPsec packets that will be reported as AH/ESP in Rx descriptor
[2] No Layer4 offload done on packets with IPSec errors
[3] no split is done for ESP packets w/ SCTP as layer4 protocol
[4] No Layer4 offload done on ESP packets with ICV error
[5] no split is done for ESP packets w/ SCTP as layer4 protocol

7.9 Virtualization

7.9.1 Overview

I/O virtualization is a mechanism to share I/O resources among several consumers. For example, in a virtual system, multiple operating system are loaded and each executes as though the whole system's resources were at its disposal. However, for the limited number of I/O devices, this presents a problem because each operating system might be in a separate memory domain and all the data movement and device management has to be done by a VMM (Virtual Machine Monitor). VMM access adds latency and delay to I/O accesses and degrades I/O performance. Virtualized devices are designed to reduce the burden of VMM by making certain functions of an I/O device shared and thus can be accessed directly from each guest operating system or Virtual Machine (VM).

The 82576 supports two modes of operations of Virtualized environments:

1. Direct assignment of part of the port resources to different guest OSes using the PCI sig SR IOV standard. Also known as “Native mode” or pass through mode. This mode is referenced as IOV mode through this chapter
2. Central management of the networking resources by an IOVM or by the VMM. Also known as software switch acceleration mode. This mode is referenced as Next Generation VMDq mode.

In a virtualized environment, the 82576 serves up to 8 virtual machines (VMs).

The virtualization off loads capabilities provided by the 82576 apart from the replication of functions defined in the PCI-sig IOV spec are also part of Next Generation VMDq.

An hybrid model, where part of the virtual machines are assigned a dedicated share of the port and the other ones are serviced by an IOVM should also be supported. However, in this case the offloads provided to the software switch might be more limited. This model can be used when parts of the VMs runs OSes for which VF drivers are available and thus can benefit from IOV and others runs older OSes for which VF drivers are not available and are serviced by an IOVM. In this case, the IOVM is assigned one VF and receives all the packets with MAC addresses of the VMs behind it.

The following section describes the support the 82576 provides for these modes. This chapter assumes a single root implementation of IOV and no support for multi root.

7.9.1.1 Direct Assignment Model

The direct assignment support in the 82576 is built according to the following model of the software environment.

It is assumed that one of the software drivers sharing the port hardware behaves as a master driver (Physical Function or PF driver). This driver is responsible for the initialization and the handling of the common resources of the port. All the other drivers might read part of the status of the common parts but can not change it. The PF driver might run either in the VMM or in some service operating system. It might be part of an IOVM or part of a dedicated service operating system.

In addition, part of the non time critical tasks are also handled by the PF driver. For example, access to CSR through the I/O space or access to the configuration space are available only through the master interface. Time critical CSR space like control of the Tx and Rx queue or interrupt handling is replicated per VF, and directly accessible by the VF driver.

Note: In some systems with a Thick Hypervisor, the Service operating system might be an integral part of the VMM - for these systems, each reference to the service operating system in the document below refers to the VMM.

7.9.1.1.1 Rationale

Direct assignment purpose is to enable each of the virtual machines to receive and transmit packets with minimum of overhead. The non time critical operations such as init and error handling can be done via the PF driver. In addition, it is important that the VMs can operate independently with minimal disturbance. It is also preferable that the VM interface to the hardware should be as close as possible to the native interface in non virtualized systems in order to minimize the software development effort.

The main time critical operations that require direct handling by the VM are:

1. Maintenance of the data buffers and descriptor rings in host memory. In order to support this, the DMA accesses of the queues associated to a VM should be identified as such on the PCIe using a different requester ID.
2. Handling of the hardware ring (tail bump and head updates)
3. Interrupts handling.

The capabilities needed to provide independence between VMs are:

1. Per VM reset and enable capabilities.
2. TX Rate control
3. Allocation of separate CSR space per VM. This CSR space is organized as close as possible to the regular CSR space to allow sharing of the base driver code.

Note: The rate control and VF enable capabilities are controlled by the PF.

7.9.1.2 System Overview

The following drawings describe the various elements involved in the I/O process in a virtualized system. Figure 7-24 describes the flow in a software Next Generation VMDq mode and Figure 7-25 the flow in IOV mode.

This document assumes that in IOV mode, the driver on the guest operating system is aware that it works in a virtual system (para-virtualized) and there is a channel between each of the virtual machine drivers and the PF driver allowing message passing such as configuration request or interrupt messages. This channel might use the mailbox system implemented in the 82576 or any other mean provided by the VMM vendor.

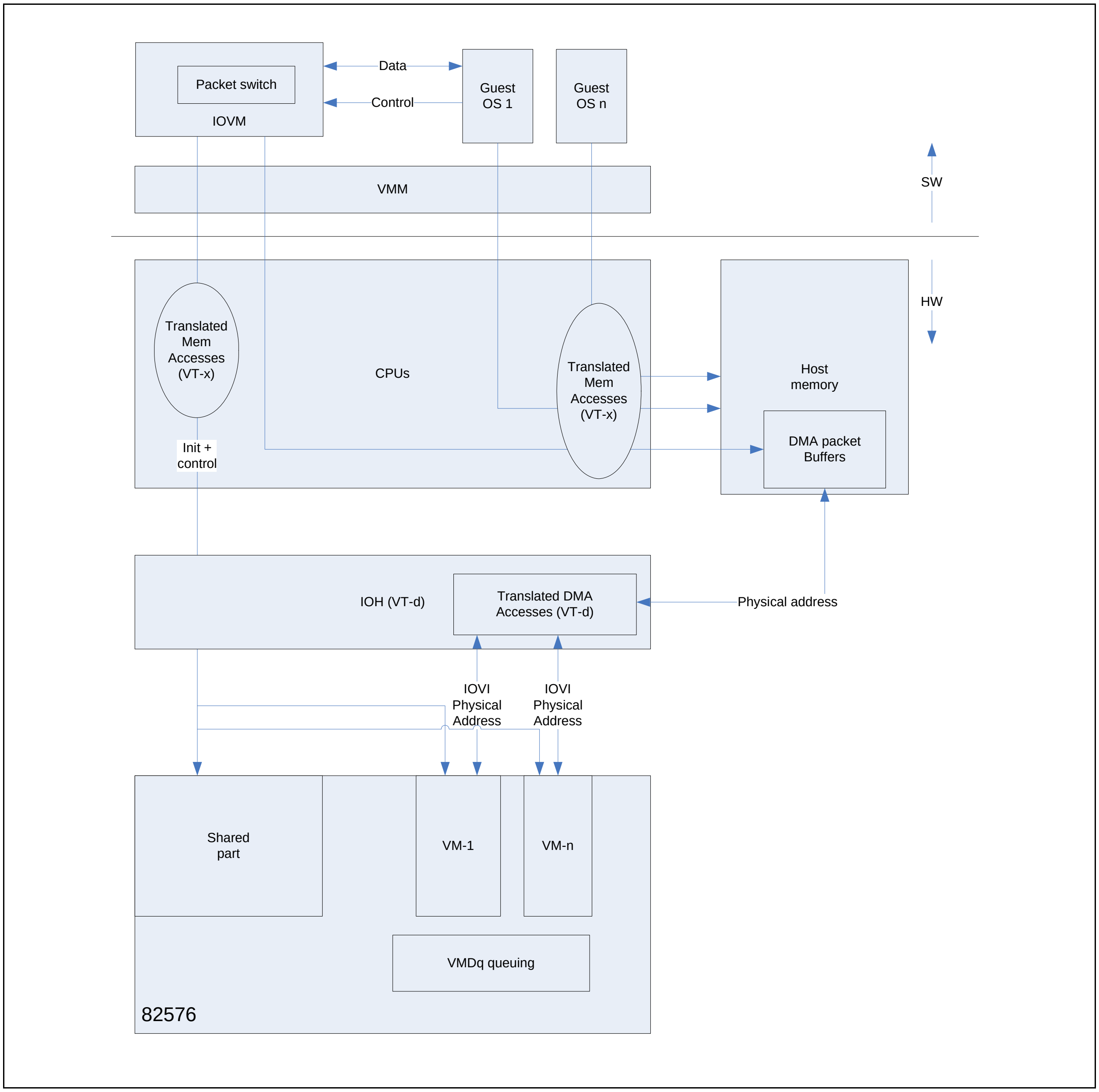


Figure 7-24. IOVM System

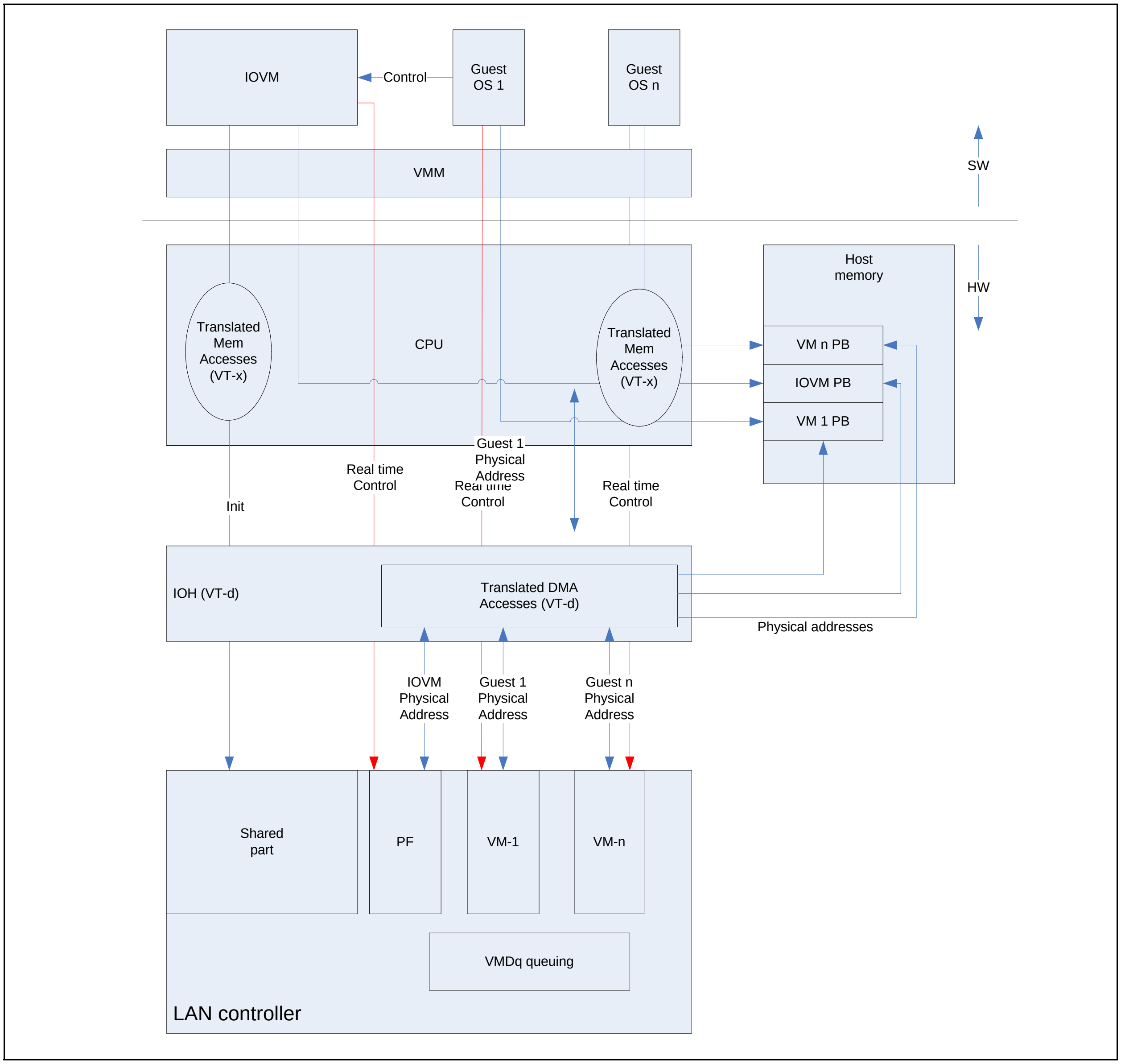


Figure 7-25. SR-IOV Based System

7.9.1.3 VMDq1 Versus Next Generation VMDq

The 82576 supports additional virtualization features not supported in 82575. The following table compares the 82576 virtualization features (Next Generation VMDq) with 82575's virtualization features (VMDq1)

Table 7-58. VMDq1 versus Next Generation VMDq

Feature	VMDq1	Next Generation VMDq
Queues	4	16
Pools	2/4	8
MAC addresses	16	24
Queuing to pool method	SA or VLAN	SA or VLAN or (SA and VLAN)
RSS in pool	Redirection table per pool.	Common redirection table - enable per pool.
VM to VM Switching	No	Yes
Broadcast and multicast Replication	No	Yes
MAC and VLAN Anti spoof protection	No	Yes
Drop if no pool	No	Yes
Per pool statistics	No	Yes
Per pool offloads	Partial	Yes
Mirroring	No	Yes
Long packet filtering	Global	Global and per pool
Storm Control	No	Yes

7.9.2 PCI Sig SR-IOV Support

7.9.2.1 SR-IOV Concepts

IOV defines the following entities in relation to I/O virtualization:

1. Virtual Image (VI): A virtual machine to which a part of the I/O resources is assigned. Also known as a VM.
2. I/O Virtual Intermediary (IOVI) or I/O Virtual Machine (IOVM): A special virtual machine that owns the physical device and is responsible for the configuration of the physical device.
3. End point (EP): The physical device that might contain a few physical functions - in our case, the 82576.
4. Physical function (PF): A function representing a Physical instance - in our case, one port. The PF driver is responsible for the configuration and management of the shared resources in the function.
5. Virtual function (VF): A part of a PF assigned to a VI.

7.9.2.2 Config Space Replication

The SR-IOV spec defines a reduced configuration space for the Virtual functions. Most of the PCIe configuration of the VFs is inherited from the PF.

This section describes the expected handling of the different parts of the configuration space for virtual functions. It deals only with the parts relevant to the 82576.

Details of the configuration space for virtual functions can be found in Section 9.7.

7.9.2.2.1 Legacy PCI Config Space

The legacy configuration space is allocated to the PF only and emulated for the VFs. A separate set of BARs and one Bus master enable bit is allocated in the SR-IOV capability structure in the PF and is used to define the address space used by the whole set of VFs.

All the legacy error reporting bits are emulated for the VF. See Section 7.9.2.4 for details.

7.9.2.2.2 Memory BARs Assignment

The SR- IOV spec defines a fixed stride for all the VF BARs, so that each VF can be allocated part of the memory BARs at a fixed stride from the a basic set of BARs. In this method only two decoders per replicated BAR per PF are required and the BARs reflected to the VF are emulated by the VMM

The only BARs that are useful for the VFs are BAR0 & BAR3, thus only those are replicated.

The following table describes the existing BARs and the stride used for the VFs:

Table 7-59. VF BARs in the 82576

BAR	Type	Usage	Requested Size Per VF
0	Mem	CSR space	min(16K, page size)
1	Mem	High word of CSR space address	N/A
2	N/A	Not used	N/A
3	Mem	MSI-X	min(16K, page size)
4	Mem	High word of MSI-X space address	N/A
5	N/A	Not used	N/A

BAR0 of the VFs are a sparse version of the original PF BAR and includes only the register relevant to the VF. For more details see Section 8.25.

The following figure describes the different BARs in an IOV enabled system:

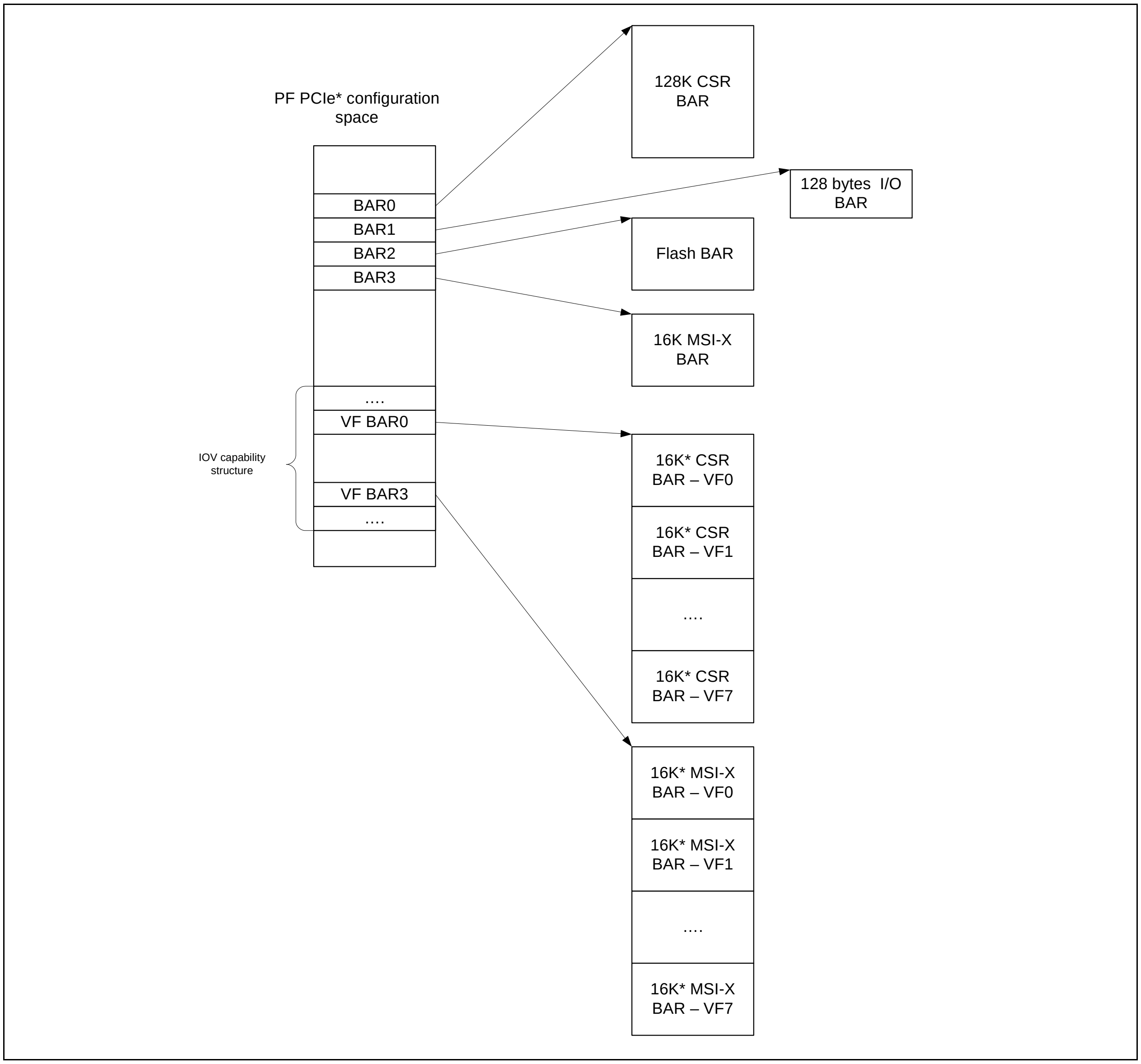


Figure 7-26. Memory BAR Allocation to VFs

7.9.2.2.3 PCIe Capability Structure

7.9.2.2.4 PCI-Express capability structure

The PCIe* capability structure is shared between the PF and the VFs. The only relevant bits that are replicated are:

1. Transaction pending
2. Function Level Reset. See Section 7.9.2.3 for details.

7.9.2.2.5 MSI and MSI-X Capabilities

Both MSI & MSI-X are implemented in the PF in the 82576. MSI-X vectors can be assigned per VF. MSI is not supported for the VFs.

See Section 9.7 for more details of the MSI-X & PBA tables implementation.

7.9.2.2.6 VPD Capability

VPD is implemented only once and is accessible only from the PF.

7.9.2.2.7 Power Management Capability

PCI SIG SR-IOV specification makes VF power management optional. The 82576 does not support power management in VFs. The power management registers are implemented in the VFs but supports only D0 state. Power management is emulated for the VFs by the IOVM and is implemented only for the PF.

7.9.2.2.8 Serial ID

Same serial ID is reported to all VFs in the device.

7.9.2.2.9 Error Reporting Capabilities (Advanced & Legacy)

All the bits in this capability structure is implemented only for the PF. The VMs see an emulated version of this. See Section 7.9.2.4 for details.

7.9.2.3 Function Level Reset (FLR) Capability

The *FLR* bit is required per VF. Setting of this bit resets only the part of the logic dedicated to the specific VF and do not influence the shared part of the port. This reset should disable the queues, disable interrupts and stop receive and transmit process per VF.

Setting the *PF FLR* bit resets the entire function.

7.9.2.4 Error Reporting

Error reporting includes baseline error reporting and AER (advanced error reporting or role based) capability.

The baseline error management includes the following functions:

1. Error capabilities enablement. These are set by the PF for all the VFs. Narrower error reporting for a given VM can be achieved by filtering of the errors by the VMM.
2. Error status in the config space. These should be set separately for each VF.
   a. See Section 9.7.1 for details about VF specific error reporting registers.

AER capability includes the following functions:

1. Error capabilities enablement. The Error Mask, and Severity bits are set by the PF for all the VFs. Narrower error reporting for a given VM can be achieved by filtering of the errors by the VMM.
2. Non-Function Specific Errors Status in the config space.
   a. Non-Function Specific Errors are logged in the PF
   b. Error logged in one register only
   c. VMM avoids touching all VFs to clear device level errors
3. Function Specific Errors Status in the config space.

   a. Function Specific Errors are logged in the VF.
   b. Allows Per VF error detection and logging.
   c. Help with fault isolation.
   d. See Section 9.7.2.3 for details about VF specific AER registers.

4. Error logging. Each VF has it’s own header log.
5. Error messages. In order to ease the detection of the source of the error, the error messages should be emitted using the requester ID of the VF in which the error occurred.

7.9.2.5 ARI & IOV Capability Structures

In order to allow more than 8 functions per end point without requesting an internal switch, as usually needed in virtualization scenarios, the PCI Sig defines the Alternative Routing ID (ARI) capability structure. This is a new capability that allows an interpretation of the Device & Function fields as a single identification of a function within the bus. In addition a new structure used to support the IOV capabilities reporting and control is defined. Both structures are described in sections 9.6.3 & 9.6.4. See next section for details on the RID allocation to VFs.

7.9.2.6 Requester ID Allocation

The requester ID allocation of the VF is done using the *Offset* field in the IOV structure. This field should be replicated per VF and is used to do the enumeration of the VFs.

Each PF includes an offset to the first associated VF. This pointer is a relative offset to the BDF of the first VF. The *Offset* field is added to PF’s requester ID to determine requester ID of the next VF. An additional field in the IOV capability structure describes the distance between two consecutive VF’s requester ID.

7.9.2.6.1 Bus-Device-Function Layout

7.9.2.6.1.1 ARI Mode

The ARI proposal allows interpretation of the device ID part of the RID as part of the function ID inside a device. Thus a single device can span up to 256 functions. In order to ease the decoding, the least significant bit of the function number points to the physical port number. The next bits indicates the VF number. The following table describes the VF requester IDs.

Table 7-60. RID Per VF - ARI Mode

Port	VF#	B,D,F	Binary	Notes
0	PF	B,0,0	B,00000,000	PF
1	PF	B,0,1	B,00000,001	PF
0	0	B,16,0	B,10000,000	Offset to first VF from PF is 128.
1	0	B,16,1	B,10000,001	
0	1	B,16,2	B,10000,010	
1	1	B,16,3	B,10000,011	
0	2	B,16,4	B,10000,100	
1	2	B,16,5	B,10000,101	
...				
0	7	B,17,6	B,10001,110	
1	7	B,17,7	B,10001,111	Last

7.9.2.6.1.2 Non ARI Mode

When ARI is disabled, non zero devices in the first bus can not be used, thus a second bus is needed to provide enough requester IDs. In this mode, the RID layout is as follows:

Table 7-61. RID Per VF - Non ARI Mode

Port	VF#	B,D,F	Binary	Notes
0	PF	B,0,0	B,00000,000	PF
1	PF	B,0,1	B,00000,001	PF
0	0	B+1,16,0	B+1,10000,000	Offset to first VF from PF is 384.
1	0	B+1,16,1	B+1,10000,001	
0	1	B+1,16,2	B+1,10000,010	
1	1	B+1,16,3	B+1,10000,011	
0	2	B+1,16,4	B+1,10000,100	
1	2	B+1,16,5	B+1,10000,101	
...				
0	7	B+1,17,6	B+1,10001,110	
1	7	B+1,17,7	B+1,10001,111	Last

Note: When the device ID of a physical function changes (because of LAN disable or Lan function Sel settings), the VF device IDs changes accordingly.

7.9.2.7 Hardware Resources Assignment

The main resources to allocate per VM are queues and interrupts. The assignment is a static one. If a VM requires more resources, it might be allocated more than one VF. In this case, each VF gets a specific MAC address/VLAN tag in order to allow forwarding of incoming traffic. The two VFs are then teamed in software.

7.9.2.7.1 Physical Function Resources

A possible use of the Physical function is for configuration setting without transmit and receive capabilities. In this case it is not allocated any queues and is allocated one MSI-X vector.

Physical function have access to all the resources of all the virtual machines but it is not expected to make use of resources allocated to active Virtual Functions.

7.9.2.7.2 Resource Summary

The 82576 supports 8 VMs, 2 queues pairs (Tx/Rx) per VM and 3 MSI-X vectors per VM.

7.9.2.8 CSR Organization

The CSR of the NIC can be divided to three types:

1. Global Configuration registers that should be accessible only to the PF (such as link control, LED control, etc.). This type of registers includes also all the debug features such as the mapping of the packet buffers and is responsible for most of the CSR area requested by the NIC. This includes per VF configuration parameters that can be set by the PF without performance impact.
2. Per queue parameters that should be replicated per queue (head, tail, Rx buffer size, and DCA tag). These parameters are used both by a VF in an IOV system and by the PF in a non IOV mode.

3. Per VF parameters (per VF reset) interrupt enable. Multiple instances of these parameters are used only in an IOV system and only one instance is needed for non IOV systems.

In order to support IOV without distributing the current drivers operation in legacy mode, the following method is used:

1. The PF instance of BAR0 continues to contain the legacy and control registers. It is accessible only to the PF. The BAR allows access to all the resources including the VF queues and other VF parameters. However it is expected that the PF driver does not access these queues in IOV mode.
2. The VF instances of BAR0 provides the control on the VF specific registers. These BARs have the same mapping as the original BAR0 with the following exceptions:
   a. Fields related to the shared resources are reserved.
   b. The 2 queues of the VF are mapped at the same location as the 2 first queues of the PF.
3. Assuming some backward compatibility is needed for IOV drivers, The PF/VF parameters block should contain a partial register set as described in Section 8.25.

7.9.2.9 IOV Control

In order to control the IOV operation, the physical driver is provided with a set of registers. These includes:

1. The mailbox mechanism described below.
2. The switch and filtering control registers described in Section 7.9.3.11.
3. VFLRE: register indicating that a VFLR reset occurred in one of the VFs (bitmap).
4. VFTE: Enables Tx traffic per VF. A VF Tx is disabled by an FLR to this VF until the PF enables it again. This allows the PF to block transmit process until the configurations for this VM are done.
5. VFRE: Enables Rx filtering per VF. A VF Rx is disabled by an FLR to this VF until the PF enables it again. This allows the PF to block the receive process until the configurations for this VM are done.

7.9.2.9.1 VF to PF Mailbox

The VF drivers and the PF driver requires some mean of communication between them. This channel can be used for the PF driver to send status updates to the VFs (link change, memory parity error, etc.) or for the VF to send requests to the PF (add to VLAN).

Such a channel can be implemented in software, but it requires enablement by the VMM vendors. In order to avoid the need for such an enablement, the 82576 provides such a channel that allows direct communication between the two drivers.

The channel consists of a mailbox similar to the host interface currently defined between the software and the manageability firmware. Each driver can then receive and indication (either poll or interrupt) when the other side wrote a message.

Assuming a max message size of 64 bytes (one cache line), RAM of 64 bytes x 8 VMs= 0.5 Kbyte is provided. Table 7-62 shows how RAM is organized.

Table 7-62. Mailbox Memory

RAM Address	Function	PF BAR 0 Mapping[1]	VF BAR 0 Mapping[2]
0 - 63	VF0 ↔ PF	0 - 63	VF0 + MBO
64 - 127	VF1 ↔ PF	64 - 127	VF1 + MBO
....			
448 - 512	VF7 ↔ PF	448 - 512	VF7 + MBO

[1] Relative to mailbox offset

2. MBO = mailbox offset in VF CSR space

In addition for each VF, the VFMailbox & PFMailbox registers are defined in order to coordinate the transmission of the messages. These registers contains a semaphore mechanism to allow coordination of the mailbox usage.

The PF driver can decide which VFs are allowed to interrupt the PF to indicate a mailbox message using the MBVFIMR mask register.

The following flows describes the usage of the mailbox:

Table 7-63. PF to VF Messaging Flow

Step	PF Driver	Hardware	VF #n Driver
1	Set PFMailbox[n].PFU		
2		Set PFU bit if PFMailbox[n].VFU is cleared	
3	Read PFMailbox[n] and check that PFU bit was set. Otherwise wait and go to step 1		
4	Write message to relevant location in VMBMEM		
5	Set the PFMailbox[n].STS bit and wait for ACK[1].		
6		Indicate an interrupt to VF #n	
7			Read the message from VMBMEM
8			Set the VFMailbox.ACK bit
9		Indicate an interrupt to PF	
10	Clear PFMailbox[n].PFU		

[1] The PF might implement a timeout mechanism to detect non responsive VFs.

Table 7-64. VF to PF Messaging Flow

Step	PF Driver	Hardware	VF #n Driver
1			Set VFMailbox.VFU
2		Set VFU bit if VFMailbox[n].PFU is cleared	
3			Read VFMailbox[n] and check that VFU bit was set. Otherwise wait and go to step 1
4			Write message to relevant location in VMBMEM
5			Set the VFMailbox.REQ bit
6		Indicate an interrupt to PF via ICR.VMMB	
7	Read MBVFICR to detect which VF caused the interrupt		
8	Read the adequate message from VMBMEM		
9	Set the PFMailbox.ACK bit		
10		Indicate an interrupt to VF #n	
11			Clear VFMailbox.VFU

The content of the message is hardware independent and can be fixed by the software.

The messages currently assumed by this specification are:

1. Registration to VLAN/Multicast packet/Broadcast packets - A VF can request to be part of a given VLAN or to get some multicast/broadcast traffic.
2. Reception of large packet - Each VF should notify the PF driver what is the largest packet size allowed in receive.
3. Get global statistics - A VF can request information from the PF driver on the global statistics.
4. Filter allocation request - A VF can request allocation of a filter for queuing/immediate interrupt support.
5. Global interrupt indication.
6. Indication of errors.

7.9.2.10 Interrupt Handling

Interrupts can be separated into two types:

1. Interrupts relevant to the behavior of each VM, including Rx & Tx packet sent indications, mailbox message and device status indications.
2. Interrupts relevant only to the handling of the shared resources. These are mainly error indications - such as packet buffer full and parity errors.

The first type of interrupts should be provided directly to the VM driver and the second type can be handled by the PF driver.

Interrupt control in the VF uses the same mechanism as the in the non virtualized case. The cause bits are independent and each VF can clear its own cause bits independently. The following registers are added per VF:

1. VTEICR, VTEICS, VTEIMS, VTEIMC, VTEIAC, VTEIAM with the following fields:
   a. RTxQ[1:0]
   b. Mailbox
2. VTEITR0,1,2.
3. VTIVAR & VTIVAR_MISC for Mailbox.

7.9.2.10.1 Low latency Interrupts

Low Latency Interrupts (LLI) are described in Section 7.3.6. Several packet types generate LLI:

- A packet matching a 5-tuple filter assigned to a VF - Each VF can require from the PF driver an LLI for one of its flows.
- A packet matching a L2 EtherType filter - an LLI is generated to specific VFs (based on the queue assignment) that handle control traffic.
- A packet matching a certain VLAN priority - an LLI is generated to the target VF based on the queue assignment for the Rx packet

An AND condition on the VM number is added to the immediate interrupt decision in order to prevent a VM from requiring immediate interrupts for flows not owned by it and in order to allow a filter to apply only to a given VM. For example, assume a given VM would require immediate interrupts on packets with PSH flag set. The VM number filtering prevents other VM from receiving immediate interrupts on such packets.

7.9.2.10.2 MSI-X

MSI-X tables are in BAR3. The MSI-X vectors might be used either as one big set of vectors in non IOV mode or as small sets allocated to VFs. In order to support both modes and save the need for duplication of the logic the first IOV vectors should be mapped as non IOV vectors also. The mapping of the vectors in IOV mode is described in Section 8.25.3

The PBA vector is replicated for the IOV case, as the saving in area is low and different per bit encoding is complicated.

7.9.2.10.3 MSI

MSI implementation is optional in the IOV spec. The 82576 doesn't support MSI in Virtual functions.

7.9.2.10.4 Legacy Interrupt (INT-x)

Legacy interrupts are not supported in IOV mode.

7.9.2.11 DMA

7.9.2.11.1 Requester ID

Each VF is allocated a requester ID. Each DMA request should use the RID of the VM that requested it. See Section 7.9.2.6 for details.

7.9.2.11.2 Sharing DMA Resources

The outstanding requests an completion credits are shared between all the VFs. The tags attached to read requests are assigned the same way they are today, although in VF systems tags can be re-used for different requester IDs.

7.9.2.11.3 DCA

The DCA enable is common to all the devices (all PFs & VFs). Given a DCA enabled device, each VM might decide for each queue, on which type of traffic (data, headers, Tx descriptors, Rx descriptors) DCA should be asserted and what is the CPU ID assigned to this queue.

Note: There are no plans to virtualize DCA in the IOH. Thus the physical CPU ID should be used in the programming of the *CPUID* field.

7.9.2.12 Timers and Watchdog

7.9.2.12.1 TCP Timer

The TCP timer is available only to the PF. It might indicate an interrupt to the VFs via the mailbox mechanism.

7.9.2.12.2 IEEE 1588

IEEE 1588 is a per link function and thus is controlled by the PF driver. The VMs have access to the real time clock register.

7.9.2.12.3 Watchdog.

The watchdog was originally developed for pass-through NICs where virtualization is not an interesting use case. Thus, this functionality is used only by the PF.

7.9.2.12.4 Free Running Timer

The free running timer is a PF driver resource the VMs can access. This register is read only to all VF. It is reset only by the PCI reset.

7.9.2.13 Power Management and Wakeup

Power management is a PF resource and is not supported per VF.

7.9.2.14 Link Control

The link is a shared resource and as such is controllable only by the PF. This include PHY settings, speed and duplex settings, flow control settings, etc. The flow control packets are sent with the station MAC address stored in the EEPROM. The watermarks of the flow control process and the time-out value are also controllable by the PF only.

Double VLAN is a network setting and as such should be common to all VFs.

7.9.2.14.1 Special Filtering Options

Pass Bad packets is a debug feature. As such pass bad packet is available only to the PF. Bad packets is passed according to the same filtering rules of the regular packets. As it might cause guest operating system s to get unexpected packets, it should be used only for debug purposes of the whole system.

Reception of long packet is controlled separately per VM. As this impact the flow control thresholds, the PF should be made aware of the decision of all the VMs. Because of this, the setup of the large send packets is centralized by the PF and each VF might request this setting.

7.9.2.14.2 Allocation of memory space for IOV functions

If the BIOS didn’t allocate memory for the IOV functions, the following flow may be used to allocate memory to the 82576 virtual function:

1. In the EEPROM request some space for the serial flash BAR. This space should be large enough to cover the IOV VF memory space needs. For example, assuming the memory page size is 4K and 8 VFs are enabled, then 256 KBytes of RAM should be requested (16 K for the CSR BAR, 16 K for the MSI-X BAR by 8 functions).
2. Before enabling IOV, zero the flash BAR and program the IOV BARs to use the old flash BAR. The VFs CSR BAR may use the first half of the original flash memory and the MSI-X BAR may use the second half.

7.9.3 Packet Switching

7.9.3.1 Assumptions

The following assumption are made:

1. The required bandwidth for the VM to VM loopback traffic is low. For example, the PCIe BW is not congested by the combination of the VM to VM and the regular incoming traffic. This case is handled but not optimized for. Unless specified otherwise, Tx and Rx packets should not be dropped or lost due to congestion caused by loopback traffic.
2. Most of the offloads provided on Rx traffic are not provided for the VM to VM loopback traffic.
3. If the buffer allocated for the VM to VM loopback traffic is full, it is OK to back pressure the transmit traffic. This mean that the outgoing traffic might be blocked if the loopback traffic is congested.

4. The decision on VM to VM loopback traffic is done only according to the Ethernet DA address and the VLAN tag. There is no filtering according to other parameters (IP, L4, etc.). This switch have no learning capabilities.
5. The forwarding decisions are based on the receive filtering programming.
6. When the link is down or during flow control events, the Tx flow is stopped, and thus the local switching traffic is stopped also.

7.9.3.2 VF Selection

The VF selection is done by MAC address and VLAN tag. Broadcast & Multicast packets are forwarded according to the individual setting of each VF and might be replicated to multiple VFs.

7.9.3.2.1 Filtering Capabilities

The following capabilities exists in to decide what is the final destination of each packet in addition to the regular L2 filtering capabilities:

- 24 MAC addresses filters (RAH/RAL registers) for both unicast and multicast filtering. These are shared with L2 filtering. For example, the same MAC addresses are used to determine if a packet is received by the switch and to determine the forwarding destination.
- 32 Shared VLAN filters (VLVF registers) - each VM can be made member of each VLAN.
- Multicast exact filtering using the existing remaining RAH/RAL registers otherwise an imperfect multicast table shared between VFs.
- 256 hash filtering of multicast addresses shared between the VFs (MTA table).
- Promiscuous unicast, multicast & enable broadcast per VF.

Note: Packets for which no queueing decision was done and still accepted by the L2 filtering, is directed to the queue pool of the default VF or dropped.

7.9.3.3 L2 Filtering

L2 filtering is the 1st stage in 3 stages that determine the destination of a received packet. The 3 stages are defined in Section 7.1.1.

All received packets passes the same filtering as in the non virtualized case; regular VLAN filtering using the global VLAN table (VTA) of the PF and filtering according to the RAH/RAL registers and according to the various promiscuous bits.

Note: Every VLAN tag set in the VLVF registers should be asserted also in the VTA table.

The *RCTL.UPE* bit (Promiscuous unicast) is not available per VF and might be modified only by the PF driver.

7.9.3.4 RX Packets Switching

Rx packet switching is the 2nd stage in 3 stages that determine the destination of a received packet. The 3 stages are defined in Section 7.1.1.

As far as switching is concerned, it doesn't matter whether our virtual environment operates in IOV mode or in Next Generation VMDq mode. The VF is identified by the “pool list” as described in Section 7.9.3.4.1 and Section 7.9.3.4.2.

When working in a virtualized environment, a single Rx queue can still be determined by the Ethertype filters. If these filters don't match, then a pool list should be found. Then the switch should determine which of the 2 queues of the targeted VMs is addressed.This 3rd stage that determines the queue in the pool is described in Section 7.1.1.2.

When working in replication mode, broadcast and multicast packets can be forwarded to ore than one VM, and is replicated to more than one Rx queue. Replication is enabled by the *Rpl_En* bit in the VT_CTL register.

In virtualization modes, the pool list is a list of one or more VMs to which the packet should be forwarded. The pool list is used in choosing the target queue list except for cases in which high priority filters with take precedence. There is a difference in the way the pool list is found when replication mode is enabled or disabled.

7.9.3.4.1 Replication Mode Enabled

When replication mode is enabled, each broadcast/multicast packet can go to more than one pool. Finding the pool list should be done according to the following steps:

1. Exact unicast or multicast match - If there is a match in one of the exact filters (RAL/RAH), for unicast or multicast packets, take the *RAH.POOLSEL[7:0]* field as a candidate for the pool list.
2. Broadcast - If the packet is a broadcast packet, add pools for which their *VMOLR.BAM* bit (Broadcast Accept Mode) is set.
3. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no pools, check it against the Unicast hash table (UTA). If there is a match, add pools for which their *VMOLR.ROPE* bit (Receive Overflow packet enable) is set.
4. Multicast hash - If the packet is a multicast packet and the prior steps yielded no pools, check it against the multicast hash table (MTA). If there is a match, add pools for which their *VMOLR.ROMPE* bit (Receive Multicast packet enable) is set.
5. Multicast Promiscuous - If the packet is a multicast packet, take the candidate list from prior steps and add pools for which their *VMOLR.MPE* bit (Multicast Promiscuous Enable) is set.
6. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous steps are ignored and a full pool list is assumed for the next step.
7. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group.
   a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]*.
   b. Untagged packets: enable only pools with their VMOLR.AUPE bit set
   c. If there is no match, the pool list should be empty.

Note: In a VLAN network, untagged packets are not expected. Such packets received by the switch should be dropped, unless their destination is a virtual port set to receive these packets. The setting is done through the *VMOLR.AUPE* bit. It is assumed that VMs for which this bit is set are members of a default VLAN and thus only MAC queuing is done on these packets.

8. Default Pool- If the pool list is empty at this stage and the *VT_CTL.Dis_Def_Pool* bit is not set, then set the default pool bit in the target pool list (from VT_CTL.Def_PL).
9. Ethertype filters - If the one of the Ethertype filters (ETQF) is matched by the packet and queuing action is requested, the VM list is set to the pool pointed by the filter.
10. VFRE - If any bit in the VFRE register is cleared, clear the respective bit in the pool list.
11. Mirroring - For each of the 4 mirroring rules add the destination (mirroring) pool (VMRCTL.MP) to the pool list according to the following rules:
    a. Pool mirroring - if VMRCTL.VPME is set and one of the bits in the pool list matches one of the bits in the VMRVM register.
    b. VLAN port mirroring - if VMRCTL.VLME is set and the index of the VLAN of the packet in the VLVF table matches one of the bits in the VMRVLAN register.
    c. Uplink port mirroring - if VMRCTL.UPME is set and the pool list is not empty and the packet came from the LAN.

d. Downlink port mirroring - if VMRCTL.DPME is set and the packet came from the host and is transmitted to the network (relevant only for VM to VM traffic).

e. VFRE - If any bit in the VFRE register is cleared, clear the respective bit in the pool list.

12. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.

The above process, up to stage 9. can be logically described by the following scheme:

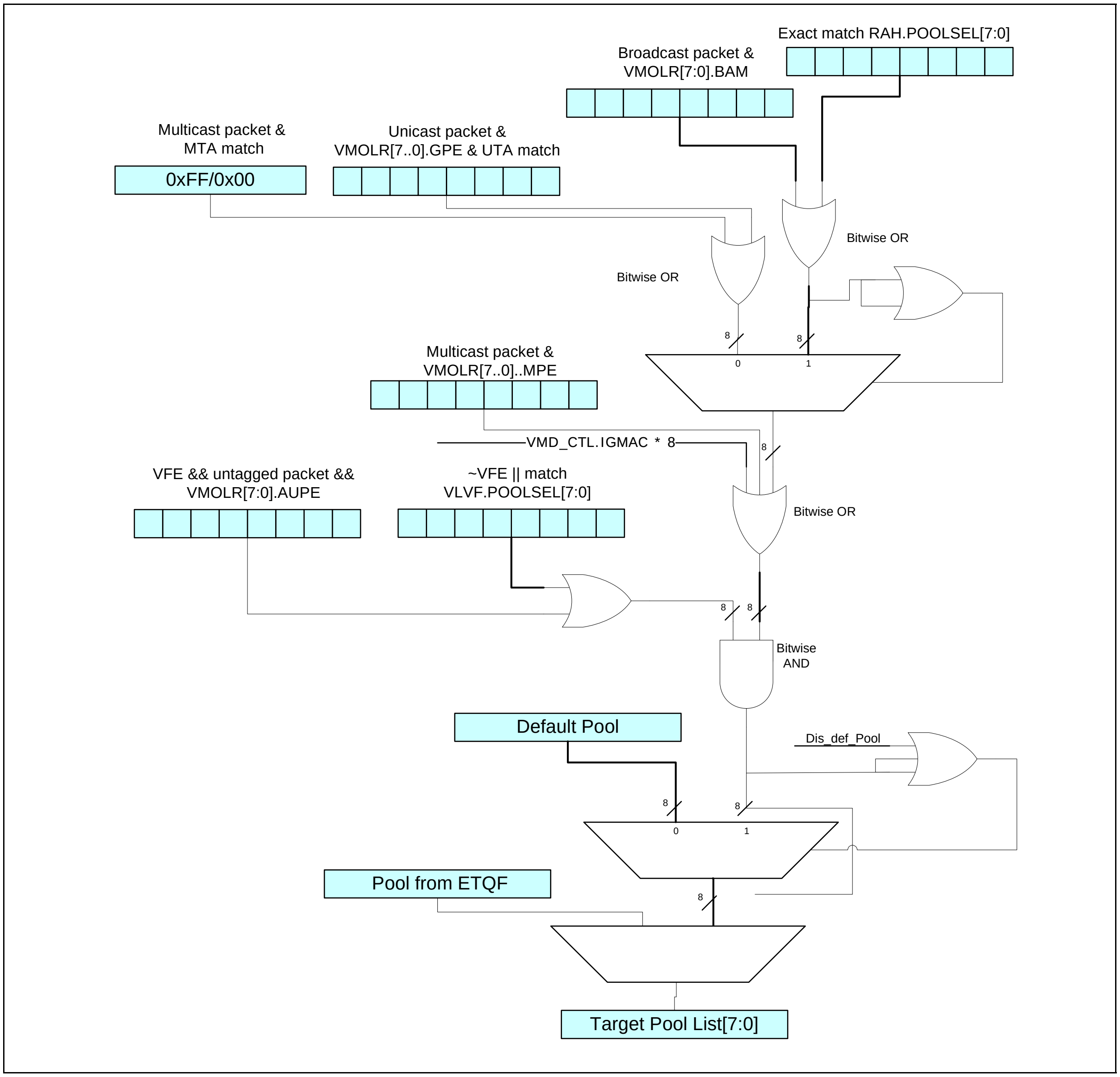

Exact match RAH.POOLSEL[7:0]
Broadcast packet & VMOLR[7:0].BAM
Multicast packet & MTA match
0xFF/0x00
Unicast packet & VMOLR[7..0].GPE & UTA match
Bitwise OR
Bitwise OR
8
8
0
1
Multicast packet & VMOLR[7..0]..MPE
VMD_CTL.IGMAC * 8
8
VFE && untagged packet && VMOLR[7:0].AUPE
~VFE || match VLVF.POOLSEL[7:0]
Bitwise OR
8
8
Bitwise AND
Default Pool
Dis_def_Pool
8
8
0
1
Pool from ETQF
8
Target Pool List[7:0]

Figure 7-27. Pool List Selection - Replication Enabled

7.9.3.4.2 Replication Mode Disabled

When replication mode is disabled, the software should take care of multicast and broadcast packets and check which of the VMs should get them. In this mode the pool list always contains one pool only according to the following steps:

1. Exact unicast or multicast match - If the packet DA matches one of the exact filters (RAL/RAH), take the *RAH.POOLSEL[7:0]* field as a candidate for the pool list.
2. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous steps are ignored and a full pool list is assumed for the next step.
3. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no pools, check it against the Unicast hash table (UTA). If there is a match, add the pool for which the *VMOLR.ROPE* bit (Receive Overflow packet enable) is set. (See software limitation no 3. below).
4. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group.
   a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]*.
   b. Untagged packets: enable only pools with their VMOLR.AUPE bit set
   c. If there is no match, the pool list should be empty.
5. Default pool- If the packet is a unicast packet and no pool was chosen and the *VT_CTL.Dis_Def_Pool* bit is not set, then set the default pool bit in the pool list (from VT_CTL.Def_PL).
6. Broadcast or Multicast - If the packet is a Multicast or Broadcast packet and was not forwarded in step 1 set the default pool bit in the pool list (from VT_CTL.Def_PL).
7. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.
8. VFRE - If any bit in the VFRE register is cleared, clear the respective bit in the pool list.

The above process can be logically described by the following scheme:

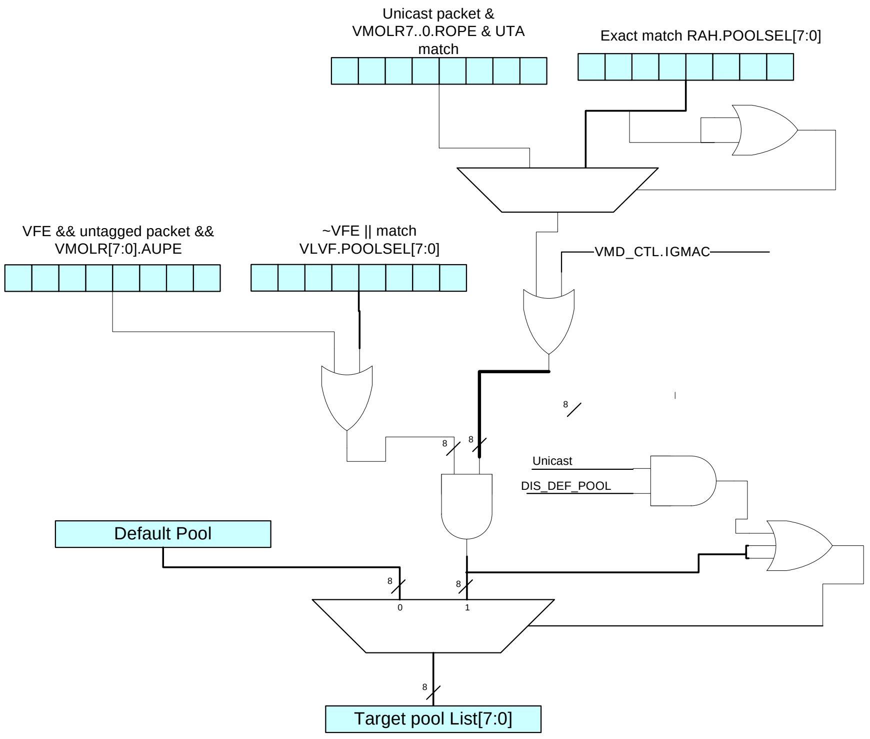


Figure 7-28. Pool List Selection - Replication Disabled

The following limitations applies when replication is disabled:

1. It is the software responsibility must not set more than one bit in the bitmaps of the exact filters. Note that multiple bits might be set in an RAH register as long as it is guaranteed that the packet is sent to only one queue by other means (VLAN)
2. The software must not set per-VM promiscuous bits (multicast or broadcast).
3. The software must not set the *Rope* bit in more than one VMOLR register.
4. If *VT_CTL.IGMAC* bit is set, the software should must not set the VMOLR.AUPE in more than one VMOLR register and must not set more than one bit in each of the VLVF.POOLSEL bitmaps.
5. The software should not activate mirroring.
6. The software should take care not to set the *Rope* bit in more than one VMOLR register.

7.9.3.5 TX Packets Switching

TX switching is used only in a virtualized environment to serve VM to VM traffic. Packets that are destined to one or more local VMs, are loop backed to the RX path through a separate packet buffer. Enabling TX switching is done by setting the *DTXSWC.Loopback_en* bit.

TX switching rules are very similar to RX switching in a virtualized environment, with the following exceptions and rules:

- There high priority filters (EType/SYN/5-Tuple) are not applied to the Tx traffic.

- If a target pool is not found, the default pool is not used, and the packet might only go to the external LAN.
- RSS is not used for queue selection inside a VM.
- A unicast packet that matches an exact filter is not sent to the LAN.
- Broadcast and multicast packets are always sent to the external LAN too, unless member of a local VLAN.
- If an outgoing packets VLAN matches a VLVF entry with the *LVLAN* bit set, this packet is not sent to the external LAN. This rule overrides previous rules.
- A packet might not be sent back to the originating VM (even if the destination address is equal to the source address). However, In order to off-load a software switch allowing Multiple VMs sharing the same pool or for VF loopback diagnostics, the 82576 provides the capability to loopback packets inside a pool. In the normal case, a packet whose source and destination are the same is dropped (usually occurs with multicast packets). If the Local Loopback bit mode (LLE) in DTXSWC is set for this pool, packets originating from a given pool can be sent to the same pool.

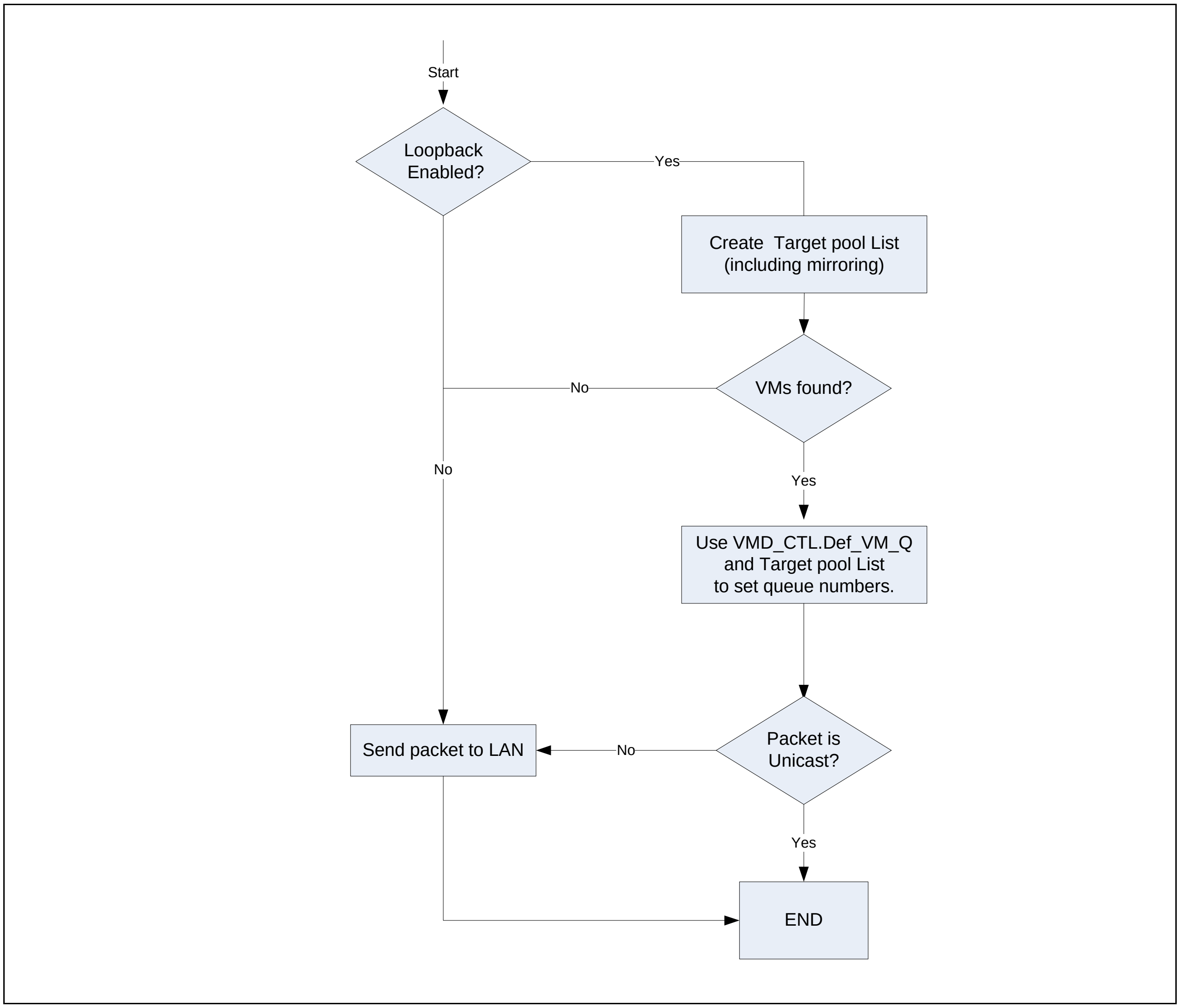

Start
Loopback Enabled?
Yes
Create Target pool List (including mirroring)
No
VMs found?
No
Yes
Use VMD_CTL.Def_VM_Q and Target pool List to set queue numbers.
Send packet to LAN
No
Packet is Unicast?
Yes
END

Figure 7-29. Tx Filtering

The following rules apply to loopback traffic:

- Loopback is disabled when the network link is disconnected. It is expected (but not required) that system software (including virtual machines) does not post packets for transmission when the link is disconnected.
- Loopback is disabled when the *Receive Enable (RXEN)* bit is cleared.
- Loopback packets are identified by the *LB* bit in the receive descriptor.

7.9.3.5.1 Replication Mode Enabled

When replication mode is enabled, the pool list for Tx packets is determined according to the following steps:

1. Exact unicast or multicast match - If there is a match in one of the exact filters (RAL/RAH), for unicast or multicast packets, take the *RAH.PLSEL* field as a candidate for the pool list.
2. Broadcast - If the packet is a broadcast packet, add pools for which their *VMOLR.BAM* bit (Broadcast Accept Mode) is set.
3. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no pools, check it against the Unicast hash table (UTA). If there is a match, add pools for which their *VMOLR.ROPE* bit (Receive Overflow packet enable) is set.
4. Multicast hash - If the packet is a multicast packet and the prior steps yielded no pools, check it against the multicast hash table (MTA). If there is a match, add pools for which their *VMOLR.ROMPE* bit (Receive Multicast packet enable) is set.
5. Multicast Promiscuous - If the packet is a multicast packet, take the candidate list from prior steps and add pools for which their *VMOLR.MPE* bit (Multicast Promiscuous Enable) is set.
6. Filter source port - The pool from which the packet was sent is removed from the pool list unless the *DTXSWC.LLE* bit is set.
7. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group.
    a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]*.
    b. Untagged packets: enable only pools with their VMOLR.AUPE bit set
    c. If there is no match, the pool list should be empty.
8. Forwarding to the Network:
    a. All broadcast and multicast packets are sent to the network also.
    b. Unicast packet that do not match any exact filter.

Note: For packets forwarded to the network, if the VLVF.LVLAN is set, then the packet is not sent to the network.

9. VFRE - If any bit in the VFRE register is cleared, clear the respective bit in the pool list.
10. Mirroring - Each of the 4 mirroring rules adds its destination pool (VMRCTL.MP) to the pool list if the following applies:
    a. Pool mirroring - VMRCTL.VPME is set and one of the bits in the pool list matches one of the bits in the VMRVM register.
    b. VLAN port mirroring - VMRCTL.VLME is set and the index of the VLAN of the packet in the VLVF table matches one of the bits in the VMVLAN register.
    c. Downlink port mirroring - VMRCTL.DPME is set and the packet is sent to the network.
    d. VFRE - If any bit in the VFRE register is cleared, clear the respective bit in the pool list.

11. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.

7.9.3.5.2 Replication Mode Disabled

When replication mode is disabled, the software should take care of multicast and broadcast packets and check which of the VMs should get them. In this mode the pool list for Tx packets always contains at the most one pool according to the following steps:

1. Exact unicast or multicast match - If the packet DA matches one of the exact filters (RAL/RAH), take the *RAH.PLSEL* field as a candidate for the pool list.
2. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no VMs, check it against the Unicast hash table (UTA). If there is a match, add pools for which their *VMOLR.ROPE* bit (Receive Overflow packet enable) is set.
3. Multicast or Broadcast - If the packet is a Multicast or Broadcast packet and was not forwarded in step 1 and 2, set the default pool bit in the pool list (from VT_CTL.DEF_PL).
4. Filter source port - The pool from which the packet was sent is removed from the pool list unless the *DTXSWC.LLE* bit is set.
5. Forwarding to the Network: All broadcast and multicast packets are sent to the network also. Unicast packet added to the pool list at step 3 (unicast hash) or for which the pool list is empty are forwarded to the network also.
6. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous steps are ignored and a full pool list is assumed for the next step.
7. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group.
   a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]*.
   b. Untagged packets: enable only pools with their VMOLR.AUPE bit set
   c. If there is no match, the pool list should be empty.
8. Forwarding to the Network:
   a. All broadcast and multicast packets are sent to the network also.
   b. Unicast packet that do not match any exact filter.

Note: For packets forwarded to the network, if the VLVF.LVLAN is set, then the packet is not sent to the network.

9. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.
10. VFRE - If any bit in the VFRE register is cleared, clear the respective bit in the pool list.

The limitations listed in Section 7.9.3.4.2 applies for Tx traffic also.

7.9.3.6 Mirroring Support

The 82576 supports 4 mirroring rules. Each rule can be of one of 5 types. Only Egress mirroring is supported and not ingress. For example, the mirroring is done on the receive path and mirrored packets reflects all the changes that occurs to the received packet. Mirroring is supported only to virtual ports and not to the uplink.

Mirroring should be activated only when one of the Next Generation VMDq queueing mode is used.

The following types of rules are supported:

1. Virtual port mirroring - reflects all the packets sent to a set of given VMs.
2. Uplink port mirroring - reflects all the traffic received from the network.
3. Downlink port mirroring - reflects all the traffic transmitted to the network.
4. Receive mirroring - reflects all the traffic received by any of the VMs. Either from the network or from local VMs. This is supported by enabling mirroring of all VMs.
5. VLAN mirroring - reflects all the traffic received in a set of given VLANs. Either from the network or from local VMs.

All the modes can be accumulated into a single rule.

This new mirroring mode is controlled by a set of rule control registers:

- VMRCTL - controls the rules to be applied and the destination port.
- VMRVLAN - controls the VLAN ports as listed in the VLVF table taking part in the VLAN mirror rule.
- VMRVM - controls the VMs ports taking part of the Virtual port mirror rule.

Mirroring is supported only when replication is enabled. The exact flow of mirroring is described in step 11. in Section 7.9.3.4.1.

7.9.3.7 Offloads

In case of packets directed to one VM only, the off loads are determined by this specific VM setting. However, the 82576 can not apply different off loads (VLAN & CRC strip + decision of size of header for split/replication offload) to different replication of the same packet. The following sections describes the rules used to decide which off loads to apply in case of replicated packets.

If replication is disabled, the offloads are determined by the unique destination of the packet.

Note: In a virtualization environment (MRQC.Multiple Receive Queues Enable = 011b - 101b), the global VLAN strip and CRC strip bits (CTRL.VME & RCTL.SECRC) are ignored and the VF specific bits in VMOLR & RPLOLR are used instead.

7.9.3.7.1 Replication by Exact MAC Address

As mentioned above, the same MAC address can be assigned to more than one VM. This is used for the following cases:

- Multicast address - In this case, the different VMs might be part of the same VLAN. The offloads applied to packets matching this address are defined in the replicated packets offloads register.
- Unicast - Same MAC different VLAN - In this case, each VM should belong to different VLAN(s). The applied offloads is according to the pool selected by the MAC/VLAN pair. One exception is the VLAN strip decision which is done according to the first pool parameters. This means that all the pools sharing a MAC address should use a common VLAN strip policy.

7.9.3.7.2 Replication by Promiscuous Modes

A packet might be replicated to multiple VMs because part of the VMs are set to receive all multicast or broadcast packets or because of a packet matching one of the hash tables (UTA or MTA).

The offloads applied to packet are defined in the replicated packets offloads registers.

In case of unicast packet, the offloads is applied according the first of the pools selected to receive the packet.

7.9.3.7.3 Replication by Mirroring

Offloads of mirrored packets are determined according to the original pool.

7.9.3.7.4 VLAN Only Filtering

If *VT_CTL.IGMAC* bit is set, the pool is defined according to the VLAN only. In this mode, only a uniform VLAN strip policy is supported. This mean the *VMOLR.STRVLAN* bit should be set to the same value for all VFs.

7.9.3.7.5 Local Traffic Offload

1. Most of the offloads available for regular incoming traffic is not available in case of VM to VM traffic. The driver might handle the lack of the offloads as follow:**VLAN strip** - A loop back tagged packet is always received by the destination VF with the VLAN tag stripped. The VP & VPKT bits in the receive descriptor indicates this conditions. The VLAN tag is received in the descriptor.
2. **Checksum** - The transmit path always adds a checksum - either by the driver of by the 82576, but this checksum is not validated by the receive path. As this packet wasn't sent over the network, the receive side might assume the TCP and IP checksum are valid.
3. **Packet types identification** - The L3 packet type identification is provided only if at least one of the following offload is requested for the transmitted packet: IP checksum, L4 checksum or IPsec offload. The L4 packet type identification is provided only if L4 checksum is requested for the transmitted packet. A packet might be identified as IPv4 with extensions only if IP checksum was requested on this packet. L5 Packet type identification is not valid for loop back packets.
4. **Header split & replication** are available only for part of the local packets. It is available only if the header split boundary is at the L4 level (TCP/UDP), in cases where the Tx side provided a valid L4 packet type (in packets for which L4 checksum is requested). In all other cases the SPH is set to zero.
5. **Error bits** - The error bits are also fixed to zero - although most of the errors are not relevant for loop back packets.
6. **RSS:** When using RSS for in pool queueing, local packets are sent to queue zero of the pool and the RSS hash is not provided.
7. **Special queueing filters** such as 5-tuple filter or ether-type filter are not applied to the local traffic. A driver using such filters should check if a packet belongs to a special queue and redirect it accordingly.

7.9.3.7.6 Small Packets Padding

In Virtualized systems, the driver receiving the packet in the VM might not be aware of all the hardware off loads applied to the packet. Thus, in case of stripping actions by the hardware (VLAN strip), it might receive packets which are smaller than a legal packet. The 82576 provides an option to pad small packets in such cases so that all packets have a legal size. This option can be enabled only if the CRC is stripped. In these cases, all packets are padded to 60 bytes (legal packet - 4 bytes CRC). The padding is done with zero data. This function is enabled via the *RCTL.PSP* bit.

7.9.3.8 Security Features

The 82576 allows some security checks on the inbound and outbound traffic of the switch.

7.9.3.8.1 Inbound Security

Each incoming packet (either from the LAN or from a local VM) is filtered according to the VLAN tag so that packets from one VLAN can not be received by VMs that are not members of that VLAN.

7.9.3.8.2 Outbound Security

7.9.3.8.2.1 Anti Spoofing

The source MAC address of each outgoing packet can be compared to the MAC address the sending VM uses for packets reception. A packet with a non matching SA is dropped. Thus preventing spoofing of the MAC address. This feature is enabled in the DTXSWC register, and can be enabled per VF.

If VLAN ant spoofing is set, a check is done to validate that sender is a member of the VLAN set in the packet. If it is not, then the packet is dropped an a notification is sent to the VMM via the *ICR.WVB* bit. This mode is controlled via the *DTXWSC.VLANAS* field.

Note: Anti Spoofing is not available for VMs that hides behind them other VMs whose MAC addresses are not part of the RAH/RAL MAC address registers. In this case anti-spoofing should be done by the software switching handling these VMs.

7.9.3.8.2.2 VLAN Insertion From Register Instead of Descriptor

There are cases, where the VLAN should be inserted by the switch without intervention from the guest operating system. In Next Generation VMDq mode, where the physical driver is controlled by a trusted central entity, we can assume the software requests inserting the right tag. However, in IOV scenarios, the driver might be malicious, and thus we can not assume it uses the right VLAN tags. In order to overcome this issue, default VLAN tags are defined per VM, and a default behavior is defined. The possible behaviors are:

1. Use descriptor value - to be used in case of a trusted VM that can decide which VLAN to send. This option should be used also in case one VM is member of multiple VLANs.
2. Always insert default VLAN - this mode should be used for non trusted or non VLAN aware VMs. In this case any VLAN insertion command from the VM is ignored. If a packet is received with a VLAN, the packet should be dropped.
3. Never insert VLAN - This mode should be used in non VLAN network. In this case any VLAN insertion command from the VM is ignored. If a packet is received with a VLAN, the packet should be dropped.

Note: The VLAN insertion settings should be done before any of the queues of the VM are enabled.

7.9.3.8.2.3 Egress VLAN Filtering

Part of the VLANs used by VMM vendors are VLAN local to the virtualized server. Packets sent with a private VLAN should not forwarded to the external network. Local VLANs are indicated by setting the *LVLAN* bit in the adequate VLVF entry.

Note: A packet with a local VLAN tag whose destination is not in the server is dropped. This means that a local VLAN should be confined to one physical port and can not have member VMs connected to different ports even in the same NIC.

7.9.3.8.3 Interrupt Misbehavior of VM.

The hardware can be programmed to take some action as a result of some misbehavior of a VM. For example upon detection of a packet with a wrong source MAC address, the hardware might block the packet. These actions might hint to the fact that some VM is malicious and the VMM should remedy to the situation. In order to inform the VMM of this fact, an interrupt is added to the ICR register (wrong VM behavior bit) to indicate the occurrence of such an action. In addition the LVMMC register contains a bitmap of all the VMs against whom some action was taken. This register is clear by read. The LVMIAC register indicates what was the last misbehavior detected.

7.9.3.9 Congestion Control

7.9.3.9.1 Receive Priority

As the switch might decide to loopback packets from the transmit path to the receive path, in case the receive path is full, the transmit path might be blocked (including the traffic to the LAN). The 82576 guarantees that packets are not dropped. The PF driver might decide to program the 82576 to drop packets from receive queues without available descriptors.

In order to keep the congestion effect locality, receive traffic from the LAN have higher priority that loop back traffic. This way large loopback traffic does not impact the network.

7.9.3.9.2 Queue Arbitration and Rate Control

In order to guarantee to each VM enough bandwidth, a per VM bandwidth control mechanism is added to the 82576. Each VM gets an allocation of transmit bandwidth and is guaranteed it can transmit within the allocation.

Received packets can be either packets received from the network or loopback packets. Packets received from the network are handled before loopback packets - no matter if the packets are unicast or replicated packets.

7.9.3.9.3 Storm Control

As there is no separate path for multicast & broadcast packets, too much replicated packets might cause congestions in the data path. In order to avoid such scenarios, broadcast and multicast storm control rate limiters are added. The rate controllers defines windows and the maximal allowed number of multicast or broadcast bytes/packets per window. Once the threshold is crossed different types of policies can be applied.

7.9.3.9.3.1 Assumptions

- Only one interval size and interval counter is used for both broadcast & multicast storm control mechanisms.
- The threshold and actions for each mechanism are separate.
- The traffic used to calculate the broadcast & multicast rate is all the traffic with a local destination - either Tx or Rx.
- The storm control does not block traffic to the network.
- The basic unit of traffic counted is 64 bytes of data.

7.9.3.9.3.2 Storm Control Functionality

The time interval over which Broadcast Storm control is performed is controlled by three factors.

- SCBI register
- Port speed.
- The value in SCCRL.INTERVAL

The first two factors determine the Unit time interval as described in Table 7-65. The interval is automatically chosen internal to hardware based on port speed. The third factor (*Interval* field) determines how many of such unit intervals are considered for one Storm Control Interval.

Table 7-65. Storm Control Interval by Speed

Port Speed	MIN Time Interval	MAX Time Interval
1 Gb/s	100 µs	100 ms
100 Mb/s	1 ms	1 s
10 Mb/s	10 ms	10 s

The number of 64 bytes chunk of Broadcast or Multicast packets that are allowed in a given interval is determined by setting the BSCTRH or MSCTRH register respectively.

The 82576 supports two modes of reactions to storm event:

1. Block all packets for the moment the threshold is crossed until the end of the interval. The block is removed at the end of the interval until the threshold is crossed again. This mode is set by asserting SCCRL.MDICW (for multicast) or SCCRL.BDICW (for broadcasts). This mode is used as a rate limiter.
2. Block all packets for the moment the threshold is crossed until a full interval without threshold crossing is registered. The block is removed at the end of the interval until the threshold is crossed again. This mode is set by asserting SCCRL.MDICW and SCCRL.MDIPW (for multicast) or SCCRL.BDICW and SCCRL.BDIPW (for broadcasts). This mode is used for storm blocking.

The 82576 might consider all packets for which a queue was not found. For example, packets that passed the 1st stage of L2 filtering but didn’t pass the 2nd stage of pooling, or where sent to the default pool, as broadcast packets. This mode is activated by setting the *SCCRL.BIDU* field.

Any change in the storm control state (block or pass of multicast or broadcast packets) is indicated to the software via the ICR.SCE interrupt cause. The current state is reflected in the SCSTS register.

For diagnostic purpose only, the storm control timer and counters can be read via the SCTC, MSCCNT & BSCCNT registers.

7.9.3.10 External Switch Loopback Support

One of the long term solutions for the switching issue is a mode where an external switch would do the loopback of VF to VF traffic and the NIC is responsible for the replication of multicast packets only. In order to support this mode, the internal loopback mode should be disabled and received packets SA should be compared to the exact MAC addresses to check if the packet originated from a local source, so that the packet is not forwarded to the originator. This mode is enabled by the *VT_CTL.FLP* bit.

7.9.3.11 Switch Control

The PF driver have some control of the switch logic. The following registers are available to the PF for this purpose:

- VLVF: VLAN queuing table: A set of 32 VLAN entries with an associated per *VF* bit map allowing allocation of each VF to each of the 32 VLAN tags.

DTXSWC: DM Tx Switch control register - controls the security setting of the switch such as MAC & VLAN anti spoof filters, local loopback enable and the loopback enable mode.

QDE: Queue Drop Enable register(s): A register defining wether receive packets destined to a specific queue is dropped if no descriptor are available. This register overrides the individual SRRCTL.DROP_EN bits.

VT_CTL: VT Control register - contains the following fields:

- Replication enable - allows replication of multicast & broadcast packets - both in incoming & outgoing traffic. If this bit is cleared, Tx multicast & broadcast packets are sent only to the network and Rx multicast & broadcast packets are sent to the default VM.
- Default pool - defines where to send packets that passed L2 filtering but didn’t pass any of the queueing mechanisms.
- Default pool disable- defines whether to drop packets that passed L2 filtering but didn’t pass any of the queueing mechanisms.

VMVIR: A set of registers used to control VLAN insertion of outgoing packets.

VMOLR/RPMOLR: Defines the offloads and pool selection options for each VF and for replicated packets.

In addition the storm control mechanism is programmed as described in Section 7.9.3.9.3.2, security features are described in Section 7.9.3.8 and the rate control mechanism is programmed.

7.9.4 Virtualization of the Hardware

This section describes additional features used in both IOV & Next Generation VMDq modes.

7.9.4.1 Per Pool Statistics

Part of the statistics are by definition shared and can not be allocated to a specific VM. For example, CRC error count can not be allocated to a specific VM, as the destination of such a packet is not known if the CRC is wrong.

All the non specific statistics is handled by the PF driver in the same way it is done in non virtualized systems. A VM might require a statistic from the PF driver but might not access it directly.

The conceptual model used to gather statistics in a virtualization context is that each queue pool is considered as a virtual link and the Ethernet link is considered as the uplink of the switch. Thus any packet sent by a VM is counted in the Tx statistics, even if it was forwarded to another VM internally or was dropped by the MAC from some reason. In the same way, a replicated packet is counted in each of the VM receiving it.

The following statistics are be provided per VM:

1. Good Packet received count.
2. Good Packet transmitted count.
3. Good octets received count.
4. Good octets transmitted count.
5. Rx Packet dropped because of queue descriptors not available (per queue).
6. Multicast Packets Received Count
7. Good Packet received from local VM count.
8. Good Packet transmitted to local VM count.
9. Good octets received from local VM count.
10. Good octets transmitted to local VM count.

Note: All the per VF statistics are RO and wrap around after reaching their maximal value.

7.10 Time SYNC (IEEE1588 and 802.1AS)

7.10.1 Overview

Measurement and control applications are increasingly using distributed system technologies such as network communication, local computing, and distributed objects. Many of these applications are enhanced by having an accurate system wide sense of time achieved by having local clocks in each sensor, actuator, or other system device. Without a standardized protocol for synchronizing these clocks, it is unlikely that the benefits are realized in the multi vendor system component market. Existing protocols for clock synchronization are not optimum for these applications. For example, Network Time Protocol (NTP) targets large distributed computing systems with millisecond synchronization requirements. The 1588 standard specifically addresses the needs of measurement and control systems:

- Spatially localized

- Microsecond to sub microsecond accuracy
- Administration free
- Accessible for both high-end devices and low-cost, low-end devices

The time sync mechanism activation is possible in full duplex mode only. No limitations on the wire speed although the wire speed might affect the accuracy.

7.10.2 Flow and Hardware/Software Responsibilities

The operation of a PTP (precision time protocol) enabled network is divided into two stages, Initialization and time synchronization.

At the initialization stage every master enabled node starts by sending Sync packets that include the clock parameters of its clock. Upon reception of a Sync packet a node compares the received clock parameters to its own. If the received clock parameters are better, than available on this node, the node moves to Slave state and stops sending Sync packets. When in slave state the node continuously compares the incoming packet clock parameters to its currently chosen master. If the new clock parameters are better then the current master selection, it changes master clock source. Eventually the best master clock source is chosen. Every node has a defined Sync packet time-out interval. If no Sync packet is received from its chosen master clock source during the interval it moves back to master state and starts sending Sync packets until a new Best Master Clock (BMC) is chosen.

The time synchronization stage is different for master and slave nodes. If a node is in master state it should periodically send a Sync packet which is time stamped by hardware on the transmit path (as close as possible to the PHY). After the Sync packet a Follow_up packet is sent which includes the value of the timestamp kept from the Sync packet. In addition the master should timestamp Delay_Req packets on its RX path and return to the slave that sent it the timestamp value using a Delay_Response packet. A node in Slave state should timestamp every incoming Sync packet that is received from its selected master, software uses this value for time offset calculation. In addition it should periodically send Delay_Req packets in order to calculate the path delay from its master. Every sent Delay_Req packet sent by the slave is time stamped and kept. Using the value received from the master Delay_Response packet the slave can now calculate the path delay from the master to the slave. The synchronization protocol flow and the offset calculation are described in Figure 7-30.

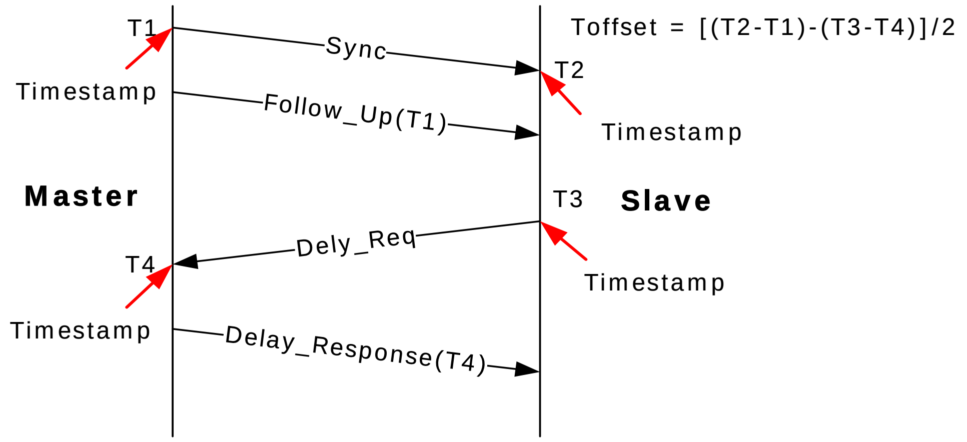


Figure 7-30. Sync Flow and Offset Calculation

The hardware's responsibilities are:

1. Identify the packets that require time stamping.
2. Time stamp the packets on both receive and transmit paths.
3. Store the time stamp value for software.
4. Keep the system time in hardware and give a time adjustment service to the software.
5. Maintain auxiliary features related to the system time.

The software's responsibilities are:

1. BMC protocol execution which means defining the node state (master or slave) and selection of the master clock if in slave state.
2. Generate PTP packets, consume PTP packets.
3. Calculate the time offset and adjust the system time using hardware mechanism.
4. Enable configuration and usage of the auxiliary features.

Table 7-66. Chronological Order of Events for Sync and Path Delay

Action	Responsibility	Node Role
Generate a Sync packet with timestamp notification in descriptor	Software	Master
Timestamp the packet and store the value in registers (T1)	Hardware	Master
Timestamp incoming Sync packet, store the value in register and store the sourceID and sequenceID in registers (T2)	Hardware	Slave
Read the timestamp from register, prepare a Follow_Up packet and send	Software	Master
Once Follow_Up packet is received, load T2 from registers and T1 from Follow_up packet	Software	Slave
Generate a Delay_Req packet with timestamp notification in descriptor	Software	Slave
Timestamp the packet and store the value in registers (T3)	Hardware	Slave
Timestamp incoming Delay_Req packet, store the value in register and store the sourceID and sequenceID in registers (T4)	Hardware	Master
Read the timestamp from register and send back to Slave using a Delay_Response packet	Software	Master
Once Delay_Response packet is received, calculate offset using T1, T2, T3 and T4 values	Software	Slave

7.10.2.1 TimeSync Indications in Receive and Transmit Packet Descriptors

Some indications need to be transferred between software and hardware regarding PTP packets.

On the receive path the hardware transfers two indications to software in the receive descriptor:

1. An indication in *RDESC.Packet Type* that this packet is a PTP packet (no matter if timestamp is sampled or not). This indication is used also by PTP packets required for protocol management.

Note: This indication is only relevant for L2 type packets (the PTP packet is identified according to its Ethertype). PTP packets have the *L2Type* bit in the *Packet Type* field set (bit 11) and the Etype matches the filter number set by the software to filter PTP packets. UDP type PTP packets don't require such an indication since the port number (319 for event and 320 for all other PTP packets) directs the packets toward the time sync application.

2. A second indication in the RDESC.STATUS.*TS* bit to indicate to the software that time stamp was taken for this packet. software needs to access the time stamp registers to get the time stamp values.

7.10.3 Hardware Time Sync Elements

All time sync hardware elements are reset to their initial values as defined in the registers section upon MAC reset.

7.10.3.1 System Time Structure and Mode of Operation

The time sync logic contains an up counter to maintain the system time value. This is a 64 bit counter that is built of the SYSTIML and SYSTIMH registers. When in Master state the SYSTIMH and SYSTIML registers should be set once by the software according to the general system requirements, when in slave state software should update the system time on every sync event as described in Section 7.10.3.3. Setting the system time is done by direct write to the SYSTIMH register and fine tune setting of the SYSTIML register using the adjustment mechanism described in Section 7.10.3.3.

Read access to the SYSTIMH and SYSTIML registers should be executed in the following order:

1. Software read register SYSTIML (at this stage the hardware latch the value of SYSTIMH).
2. Software read register SYSTIMH (the latched value (from last read from SYSTIML) is returned by the hardware).

Upon an increment event the system time value should be incremented by the value stored in TIMINCA.incvalue. Increment event happens every TIMINCA.incperiod cycles. If the TIMINCA.incperiod cycles value is one, then the increment event should occur every clock cycle. The incvalue defines the time granularity represented by the SYSTIMH/L registers. For example if the cycle time is 16 ns and the incperiod is one, then if the incvalue is 16 then the SYSTMH/L time is represented in nanoseconds. If the *incvalue* is 160 then the time is represented in 0.1ns units and so on. The *incperiod* helps to avoid inaccuracy in cases where T value can not be represented as a simple integer and should be multiplied to get to an integer representation. The *incperiod* value should be as small as possible to achieve best accuracy possible.

Note: Best accuracy is achieved at lowest permitted "Incperiod" equals to 1 and as high as possible Incvalue.

7.10.3.2 Time Stamping Mechanism

The time stamp logic is located on transmit and receive paths at a location as close as possible to the PHY, to reduce delay uncertainties originating from implementation differences. The time stamp logic operation is slightly different on transmit and on receive paths.

The transmit logic decides to timestamp a packet if the transmit timestamp is enabled (*TSYNCTXCTL.EN* = 1) and the time stamp bit in the packet descriptor (*TDESD.MAC.1588* = 1) is set. On the transmit side only the time is captured in the TXSTMPL and TXSTMPH registers.

The receive logic parses the received frame and if it is matches the *CTRLT* or *MSGT* message types defined in the TSYNCRXCFG register (see Section 8.17.23) the time, sourceId and sequenceId are latched in the RXSTMPL, RXSTMPH, RXSATRL and RXSATRH timestamp registers. In addition two indications in the receive descriptor are placed:

1. RDESC.Packet Type - Value in this field identifies that this is a PTP packet (this indication is only for L2 packets since on the UDP packets the port number directs the packet to the application).
2. RDESC.STATUS.*TS* - Bit identifies that a time stamp was taken for this packet. If this PTP packet is not Sync or Delay_Req or for some reason the time stamp was not taken the *TS* bit is not set.

For more details please refer to the timestamp registers in Section 8.16. Figure 7-31 defines the exact point where the time value is captured.

On both transmit and receive sides the timestamp values are locked in registers until values are read by software. As a result if a new PTP packet that requires time stamp arrives before software access it is not time stamped. In some cases in the receive path a packet that was timestamped might be lost and not reach the host. To avoid a deadlock condition on the time stamp registers the software should keep a watch dog timer to clear locking of the time stamp register. The interval counted by such a timer should be at least higher then the expected interval between two Sync or Delay_Req packets depends on the node state (Master or Slave).

.

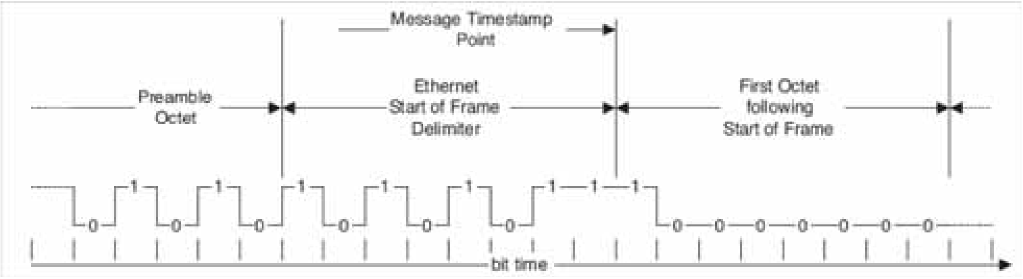


Figure 7-31. Time Stamp Point

7.10.3.3 Time Adjustment Mode of Operation

A Node in a Time Sync network can be in one of two states Master or Slave. When a Time Sync entity is in the Master state it should synchronize other entities to its System Clock. In this case no time adjustments are needed. When the entity is in slave state it should adjust its system clock by using the data arriving in the Follow_Up and Delay_Response packets and the time stamp values of the Sync and Delay_Req packets. When all the values are available the software in the slave entity can calculate its offset in the following manner:

$$Toffset = [(T2-T1) - (T3-T4)]/2$$

T1 - Timing data in Follow_Up packet
T2 - Sync Time Stamp
T3 - Delay_Req Time Stamp
T4 - Timing data in Delay_Response packet

After offset calculation the system time register should be updated. This is done by writing the calculated offset to the TIMADJL and TIMADJH registers. The order should be as follows:

1. Write the low portion of the offset to TIMADJL.
2. Write the high portion of the offset to TIMADJH to the lower 31 bits and the sign to the most significant bit.

After the write cycle to TIMADJH the value of TIMADJH and TIMADJL is added to the system time.

7.10.4 Time Sync Related Auxiliary Elements

The time sync logic implements two types of auxiliary elements using the precise system timer.

7.10.4.1 Target Time

The two target time registers *TRGTTIML/H0* and *TRGTTIML/H1* enable to generate a time triggered event to external hardware using one of the SDP pins. Each target time register is structured the same as the system time register. If the value of the system time is equal to the value written to one of the target time registers a change in level occurs on one of the SDP outputs. The accuracy of the comparison is defined by the value of the *TSAUXC.Mask* described in Section 8.17.14. Each target time register has an enable bit located in the auxiliary control register (*TSAUXC.EN_TTi*). Upon getting a target time event the enable bit is cleared and needs to be set again by software to get another target time event.

7.10.4.2 Time Stamp Events

Upon a change in the input level of one of the SDP pins that was configured to detect Time stamp events using the TSSDP register, a time stamp of the system time is captured into one of the two auxiliary time stamp registers (AUXSTMPL/H0 or AUXSTMPL/H1).

7.10.5 PTP Packet Structure

The time sync implementation supports both the 1588 V1 and V2 PTP frame formats. The V1 structure can come only as UDP payload over IPv4 while the V2 can come over L2 with its Ethertype or as a UDP payload over IPv4 or IPv6. The 802.1AS uses only the layer 2 V2 format.

Note: It is assumed that Time Sync V1 packets are never protected by IPSec.

Table 7-67. V1 and V2 PTP Message Structure

Offset in Bytes	V1 Fields	V2 Fields	
Bits	7 0	7 4	3 0
0	***versionPTP***	***transport Specific***	***messageId***
1		Reserved	***versionPTP***
2	version Network	message Length	
3			
4	Subdomain	Subdomain Number	
5		Reserved	
6		flags	
7			
8		correctionNs	
9			
10			
11			
12			
13			
14		correctionSubNs	
15			
16		reserved	
17			
18			
19			

Table 7-67. V1 and V2 PTP Message Structure (Continued)

Offset in Bytes	V1 Fields	V2 Fields	
Bits	7 0	7 4	3 0
20	message Type	Reserved	
21	Source communication technology	Source communication technology	
22	Sourceuuid	Sourceuuid	
23			
24			
25			
26			
27			
28	source port id	source port id	
29			
30	***sequenceId***	***sequenceId***	
31			
32	***control***	control	
33	reserved	log Message Period	
34	flags	N/A	
35			

Note: Only the fields with the bold italic format colored red are of interest to the hardware.

Table 7-68. PTP Message Over Layer 2

Ethernet (L2)	VLAN (Optional)	PTP Ethertype	PTP message

Table 7-69. PTP Message Over Layer 4

Ethernet (L2)	IP (L3)	UDP	PTP message

When a PTP packet is recognized (by Ethertype or UDP port address) on the receive side then if the version is V1 then the *Control* field at offset 32 should be compared to the *TSYNCRXCFG.CTRLT* message field (see Section 8.17.23) otherwise the byte at offset zero should be used for comparison to the *TSYNCRXCFG*.MSGT field. The rest of the required fields are at the same location and size for both V1 and V2.

Table 7-70. Message Decoding for V1 (Control Field at Offset 32)

Enumeration	Value
PTP_SYNC_MESSAGE	0
PTP_DELAY_REQ_MESSAGE	1
PTP_FOLLOWUP_MESSAGE	2
PTP_DELAY_RESP_MESSAGE	3
PTP_MANAGEMENT_MESSAGE	4
reserved	5–255

Table 7-71. Message Decoding for V2 (MessageId Field at Offset 0)

MessageId	Message Type	Value (hex)
PTP_SYNC_MESSAGE	Event	0
PTP_DELAY_REQ_MESSAGE	Event	1
PTP_PATH_DELAY_REQ_MESSAGE	Event	2
PTP_PATH_DELAY_RESP_MESSAGE	Event	3
Unused		4-7
PTP_FOLLOWUP_MESSAGE	General	8
PTP_DELAY_RESP_MESSAGE	General	9
PTP_PATH_DELAY_FOLLOWUP_MESSAGE	General	A
PTP_ANNOUNCE_MESSAGE	General	B
PTP_SIGNALLING_MESSAGE	General	C
PTP_MANAGEMENT_MESSAGE	General	D
Unused		E-F

If V2 mode is configured in the TSYNCRXCTL.Type field (see Section 8.17.1) then the time stamp should be taken on PTP_PATH_DELAY_REQ_MESSAGE and PTP_PATH_DELAY_RESP_MESSAGE according to the value in the *TSYNCRXCFG*.MSGT message field described in Section 8.17.23.

7.11 Statistics

The 82576 supports different statistics counters as described in Section 8.19. The statistics counters can be used to create statistics reports as required by different standards. The 82576 statistics allows support for the following standards:

- IEEE 802.3 clause 30 management – DTE section.
- NDIS 6.0 OID_GEN_STATISTICS.
- RFC 2819 – RMON Ethernet statistics group.
- Linux Kernel (version 2.6) net_device_stats

The following section describes the match between the internal the 82576 statistic counters and the counters requested by the different standards.

7.11.1 IEEE 802.3 clause 30 management

The 82576 supports the Basic and Mandatory Packages defined in clause 30 of the IEEE 802.3 spec. The following table describes the matching between the internal statistics and the counters requested by these packages.

Table 7-72. IEEE 802.3 Mandatory Package Statistics

Mandatory package capability	Intel® 82576 GbE Controller counter	Notes and limitations
FramesTransmittedOK	GPTC	The 82576 doesn’t include flow control packets.
SingleCollisionFrames	SCC	
MultipleCollisionFrames	MCC	

Table 7-72. IEEE 802.3 Mandatory Package Statistics (Continued)

FramesReceivedOK	GPRC	The 82576 doesn't include flow control packets.
FrameCheckSequenceErrors	CRCERRS	
AlignmentErrors	ALGNERRC	

In addition, part of the recommended package is also implemented as described in the following table

Table 7-73. IEEE 802.3 Recommended Package Statistics

Recommended package capability	Intel® 82576 GbE Controller counter	Notes and limitations
OctetsTransmittedOK	GOTCH/GOTCL	The 82576 counts also the DA/SA/LT/CRC as part of the octets. The 82576 doesn't count Flow control packets.
FramesWithDeferredXmissions	DC	
LateCollisions	LATECOL	
FramesAbortedDueToXSColls	ECOL	
FramesLostDueToIntMACXmitError	HTDMPC	The 82576 counts the excessive collisions in this counter, while 802.3 increments no other counters, while this counter is incremented
CarrierSenseErrors	TNCRS	The 82576 doesn't count cases of CRS de-assertion in the middle of the packet. However, such cases are not expected when the internal PHY is used.
OctetsReceivedOK	TORL+TORH	Counts also the DA/SA/LT/CRC as part of the octets. Doesn't count Flow control packets.
FramesLostDueToIntMACRcvError	RNBC	
SQETestErrors	N/A	
MACControlFramesTransmitted	N/A	
MACControlFramesReceived	N/A	
UnsupportedOpcodesReceived	FCURC	
PAUSEMACCtrlFramesTransmitted	XONTXC + XOFFTXC	
PAUSEMACCtrlFramesReceived	XONRXC + XOFFRXC	

Part of the optional package is also implemented as described in the following table

Table 7-74. IEEE 802.3 Optional Package Statistics

Optional package capability	Intel® 82576 GbE Controller counter	Notes
MulticastFramesXmittedOK	MPTC	Intel® 82576 GbE Controller doesn't count FC packets
BroadcastFramesXmittedOK	BPTC	
MulticastFramesReceivedOK	MPRC	Intel® 82576 GbE Controller doesn't count FC packets
BroadcastFramesReceivedOK	BPRC	
InRangeLengthErrors	LENERRS	
OutOfRangeLengthField	N/A	Packets parsed as Ethernet II packets
FrameTooLongErrors	ROC + RJC	

7.11.2 OID_GEN_STATISTICS

The 82576 supports the part of the OID_GEN_STATISTICS as defined by Microsoft* NDIS 6.0 spec. The following table describes the matching between the internal statistics and the counters requested by this structure.

Table 7-75. Microsoft* OID_GEN_STATISTICS

OID entry	Intel® 82576 GbE Controller counters	Notes
ifInDiscards;	CRCERRS + RLEC + RXERRC + MPC + RNBC + ALGNERRC	
ifInErrors;	CRCERRS + RLEC + RXERRC + ALGNERRC	
ifHCInOctets;	GORCL/GOTCL	
ifHCInUcastPkts;	GPRC - MPRC - BPRC	
ifHCInMulticastPkts;	MPRC	
ifHCInBroadcastPkts;	BPRC	
ifHCOutOctets;	GOTCL/GOTCH	
ifHCOutUcastPkts;	GPTC - MPTC - BPTC	
ifHCOutMulticastPkts;	MPTC	
ifHCOutBroadcastPkts;	BPTC	
ifOutErrors;	ECOL + LATECOL	
ifOutDiscards;	ECOL	
ifHCInUcastOctets;	N/A	
ifHCInMulticastOctets;	N/A	
ifHCInBroadcastOctets;	N/A	
ifHCOutUcastOctets;	N/A	
ifHCOutMulticastOctets;	N/A	
ifHCOutBroadcastOctets;	N/A	

7.11.3 RMON

The 82576 supports the part of the RMON Ethernet statistics group as defined by IETF RFC 2819. The following table describes the matching between the internal statistics and the counters requested by this group.

Table 7-76. RMON Statistics

RMON Statistic	Intel® 82576 GbE Controller Counters	Notes
etherStatsDropEvents	MPC + RNBC	
etherStatsOctets	TOTL + TOTH	
etherStatsPkts	TPR	
etherStatsBroadcastPkts	BPRC	
etherStatsMulticastPkts	MPRC	The 82576 don't count FC packets
etherStatsCRCAlignErrors	CRCERRS + ALGNERRC	
etherStatsUndersizePkts	RUC	
etherStatsOversizePkts	ROC	
etherStatsFragments	RFC	Should count bad aligned fragments as well
etherStatsJabbers	RJC	Should count bad aligned jabbers as well
etherStatsCollisions	COLC	
etherStatsPkts64Octets	PRC64	RMON counts bad packets as well

Table 7-76. RMON Statistics (Continued)

etherStatsPkts65to127Octets	PRC127	RMON counts bad packets as well
etherStatsPkts128to255Octets	PRC255	RMON counts bad packets as well
etherStatsPkts256to511Octets	PRC511	RMON counts bad packets as well
etherStatsPkts512to1023Octets	PRC1023	RMON counts bad packets as well
etherStatsPkts1024to1518Octets	PRC1522	RMON counts bad packets as well

7.11.4 Linux net_device_stats

The 82576 supports part of the net_device_stats as defined by Linux Kernel version 2.6 (defined in <linux/netdevice.h>). The following table describes the matching between the internal statistics and the counters requested by this structure.

Table 7-77. Linux net_device_stats

net_device_stats Field	Intel® 82576 GbE Controller Counters	Notes
rx_packets	GPRC	The 82576 doesn’t count flow controls - can be accounted for by using the XONRXC and XOFFRXC counters
tx_packets	GPTC	The 82576 doesn’t count flow controls - can be accounted for by using the XONTXC and XOFFTXC counters
rx_bytes	GORCL + GORCH	
tx_bytes	GOTCL + GOTCH	
rx_errors	CRCERRS + RLEC + RXERRC + ALGNERRC	
tx_errors	ECOL + LATECOL	
rx_dropped	N/A	
tx_dropped	N/A	
multicast	MPTC	
collisions	COLC	
rx_length_errors	RLEC	
rx_over_errors	N/A	
rx_crc_errors	CRCERRS	
rx_frame_errors	ALGNERRC	
rx_fifo_errors	HRMPC	
rx_missed_errors	MPC	
tx_aborted_errors	ECOL	
tx_carrier_errors	N/A	
tx_fifo_errors	N/A	
tx_heartbeat_errors	N/A	
tx_window_errors	LATECOL	
rx_compressed	N/A	
tx_compressed	N/A	

7.11.5 Statistics hierarchy.

The following diagrams describes the relations between the packet flow and the different statistic counters.

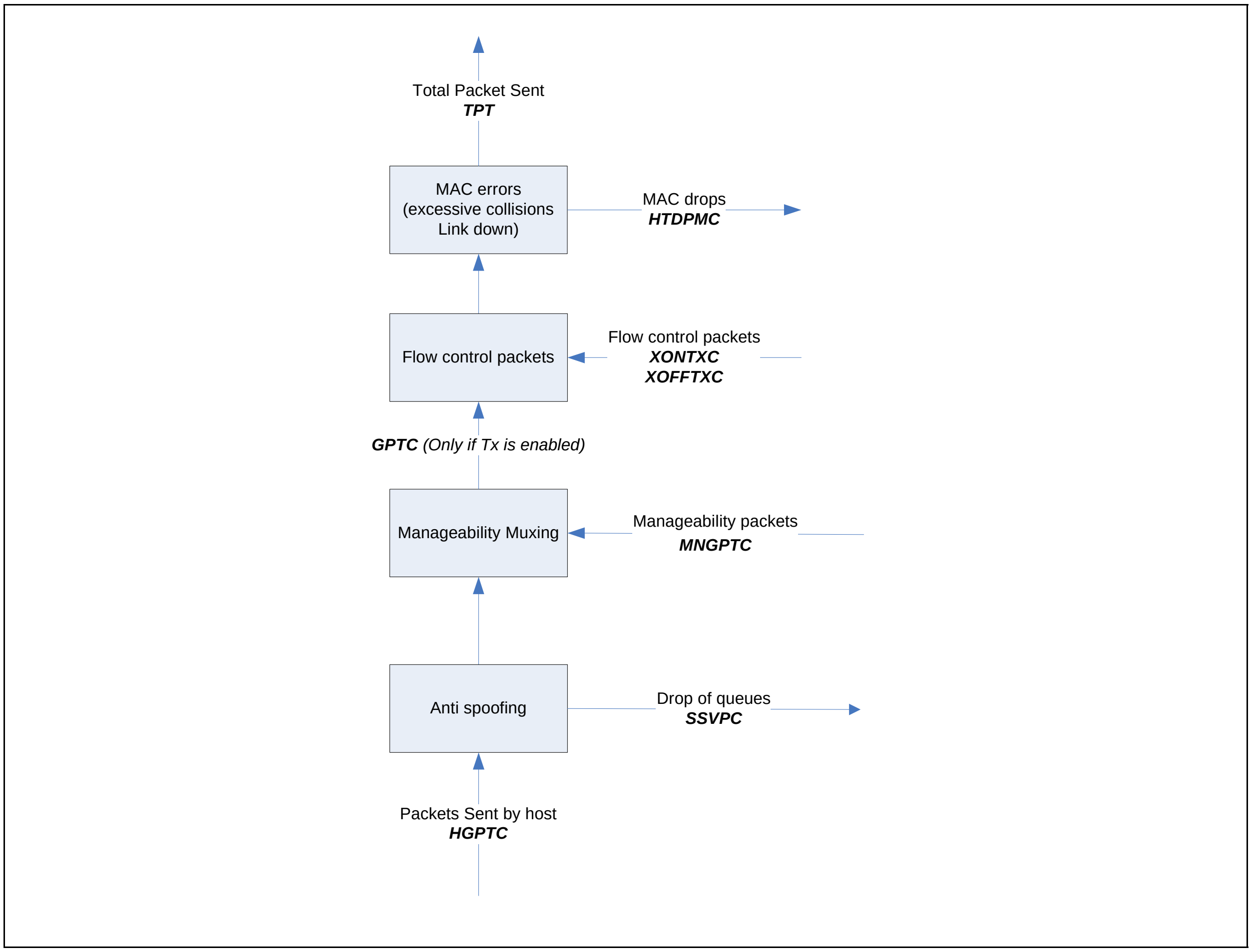


Figure 7-32. Tx Flow Statistics

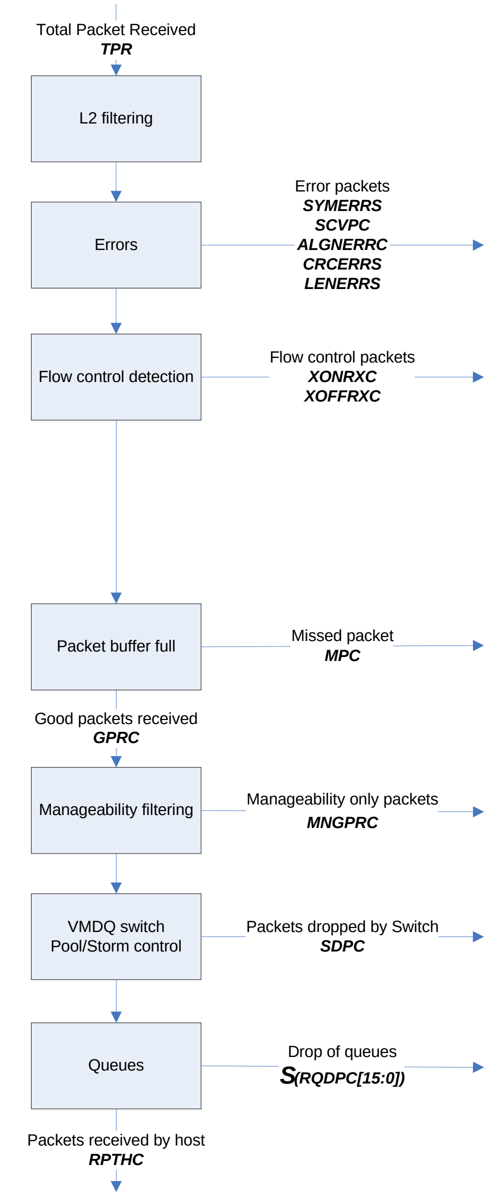


Figure 7-33. Rx Flow Statistics

§ §

NOTE: This page intentionally left blank.

8. Programming Interface

8.1 Introduction

This chapter details the programmer visible state inside the 82576. In some cases, it describes hardware structures invisible to software in order to clarify a concept. The 82576's address space is mapped into four regions with PCI Base Address Registers described in Section 9.4.11. These regions are listed in the table below.

Table 8-1. Address Space Regions

Addressable Content	How Mapped	Size of Region
Internal registers and memories	Direct memory-mapped	128K
Flash (optional)	Direct memory-mapped	64-512K
Expansion ROM (optional)	Direct memory-mapped	64-512K
Internal registers and memories, Flash (optional)	I/O Window mapped	32 bytes
MSI-X (optional)	Direct memory-mapped	16K

Both the Flash and Expansion ROM Base Address Registers map the same flash memory. The internal registers and memories and flash can be accessed though I/O space indirectly as explained below. The internal register/memory space is described in the following sections.

The PHY registers are accessed through the MDIO interface.

8.1.1 Memory and I/O Address Decoding

8.1.1.1 Memory-Mapped Access to Internal Registers and Memories

The internal registers and memories might be accessed as direct memory-mapped offsets from the base address register (BAR0 or BAR 0/1 see Section 9.4.11). See Section 8.1.3 for the appropriate offset for each specific internal register.

In IOV mode, this area is partially duplicated per VF. All replications contain only the subset of the register set that is available for VF programming.

8.1.1.2 Memory-Mapped Access to Flash

The external Flash might be accessed using direct memory-mapped offsets from the Flash base address register (BAR1 or BAR2/3 see Section 9.4.11). The Flash is only accessible if enabled thorough the EEPROM Initialization Control Word, and if the Flash Base Address register contains a valid (non-zero) base memory address. For accesses, the offset from the Flash BAR corresponds to the offset into the flash actual physical memory space.

8.1.1.3 Memory-Mapped Access to MSI-X Tables

The MSI-X tables might be accessed as direct memory-mapped offsets from the base address register (BAR3 or BAR 4/5 see Section 9.4.11). See Section for the appropriate offset for each specific internal register.

In IOV mode, this area is duplicated per VF.

8.1.1.4 Memory-Mapped Access to Expansion ROM

The external Flash might also be accessed as a memory-mapped expansion ROM. Accesses to offsets starting from the Expansion ROM Base address (see Section 9.4.11) reference the Flash provided that access is enabled thorough the EEPROM Initialization Control Word, and if the Expansion ROM Base Address register contains a valid (non-zero) base memory address.

8.1.1.5 I/O-Mapped Access to Internal Registers, Memories, 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 see Section 9.4.11), 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:

Table 8-2. IOADDR and IODATA

Offset	Ab	Name	RW	Size
0x00	IOADDR	Internal Register, Internal Memory, or Flash Location Address. 0x00000-0x1FFFF – Internal Registers and Memories 0x20000-0x7FFFF – Undefined 0x80000-0x87FFFF – Flash	RW	4 bytes
0x04	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
0x08 – 0x1F	Reserved	Reserved	RO	4 bytes

8.1.1.5.1 IOADDR (I/O offset 0x00)

The IOADDR register must always be written as a DWORD access. Writes that are less than 32 bits is ignored. Reads of any size returns a DWORD of data. However, the chipset or CPU might only return a subset of that DWORD.

For software programmers, the IN and OUT instructions must be used to cause I/O cycles to be used on the 3GIOPCIe bus. Because 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.

Because only a particular range is addressable, the upper bits of this register are hard coded to zero. Bits 31 through 20 are not write-able and always read back as 0b.

At hardware reset (Internal_Power_On_Reset) or PCI Reset, this register value resets to 0x00000000. Once written, the value is retained until the next write or reset.

8.1.1.5.2 IODATA (I/O offset 0x04)

The IODATA register must always be written as a DWORD access when the IOADDR register contains a value for the Internal Register and Memories (for example, 0x00000-0x1FFFC). In this case, writes that are less than 32 bits is ignored.

The IODATA register might be written as a BYTE, WORD, or DWORD access when the IOADDR register contains a value for the Flash (for example, 0x80000-0xFFFFF). In this case, the value in IOADDR must be properly aligned to the data value. This table identifies the supported configurations:

Table 8-3. Intel® 82576 GbE Controller IOADDR Bits

Access Type	Intel® 82576 GbE Controller IOADDR Register Bits [1:0]	Target IODATA Access BE[3:0]# bits in Data Phase
BYTE (8 bit)	00b	1110b
	01b	1101b
	10b	1011b
	11b	0111b
WORD (16 bit)	00b	1100b
	10b	0011b
DWORD (32 bit)	00b	0000b

Note: Software might have to implement non-obvious code to access the Flash at a BYTE or WORD at a time. Example code that reads a Flash byte is shown here to illustrate the impact of the above table:

```
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 might only return a subset of that DWORD.

For software programmers, the IN and OUT instructions must be used to cause I/O cycles to be used on the PCIe 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 (0x20000-0x7FFFC) should not be performed. Results cannot be determined.

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 82576 delays the results through normal bus methods (for example, split transaction or transaction retry).

Note: 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 non-deterministic from the software viewpoint.

8.1.1.5.3 Undefined I/O offsets

I/O offsets 0x08 through 0x1F are considered to be reserved offsets with the I/O window. Dword reads from these addresses returns 0xFFFF; writes to these addresses is discarded.

8.1.2 Register Conventions

All registers in the 82576 are defined to be 32 bits, should be accessed as 32 bit double-words, There are some exceptions to this rule:

- Register pairs where two 32 bit registers make up a larger logical size
- Accesses to Flash memory (via Expansion ROM space, secondary BAR space, or the I/O space) might 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 “one” by software. Reads from registers containing reserved bits might 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 to. Writing to reserved or undefined register addresses might cause indeterminate behavior. Reads from reserved or undefined configuration register addresses might 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 is listed as such via the text “undefined”, “unknown”, or “X”. Such configuration values might need to be set via EEPROM configuration or via 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 which is overridden by a higher-precedence operation. Operations which might supersede hardware defaults might include a valid EEPROM 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) does not take effect (the write is ignored). Partial reads returns all 32 bits of data regardless of the byte enables.

Notes: Partial reads to read-on-clear registers (ICR) can have unexpected results since all 32 bits are actually read regardless of the byte enables. Partial reads should not be done.

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). When reading 64 bits statistics registers the least significant 32 bit register should be read first.

See special notes for VLAN Filter Table, Multicast Table Arrays and Packet Buffer Memory which appear in the specific register definitions.

The 82576 register fields are assigned one of the attributes described in Table 8-4.

Table 8-4. Intel® 82576 GbE Controller Register Field Attributes

Attribute	Description
RW	Read-Write field: Register bits are read-write and can be either set or cleared by software to the desired state.
RWS	Read-Write Status field: Register bits are read-write and can be either set or cleared by software to the desired state. However, the value of this field might be changed by the hardware to reflect a status change.
RO	Read-only register: Register bits are read-only and cannot be altered by software. Register bits might be initialized by hardware mechanisms such as pin strapping, serial EEPROM or reflect a status of the hardware state.
R/W1C	Read-only status, Write-1-to-clear status register: Register bits indicate status when read, a set bit indicating a status event can be cleared by writing a 1b. Writing a 0b to R/W1C bit has no effect.
Rsv	Reserved.Do not write to these fields.
RC	Read-only status, Read-to-clear status register: Register bits indicate status when read, a set bit indicating a status event is cleared by reading it.
SC	Self Clear field: a command field that is self clearing. These field are always read as zero.
WO	Write only field: a command field that can not be read, These field read values are undefined.
RC/W1C	Read-only status, Write-1-to-clear status register: Read-to-clear status register Register bits indicate status when read, a set bit indicating a status event can be cleared by writing a 1b or by reading the register. Writing a 0b to RC/W1C bit has no effect.
RS	Read Set – This is the attribute used for Semaphore bits. These bits are set by read in case the previous values were zero. In this case the read value is zero; otherwise the read value is one. Cleared by write zero.

PHY registers described in Section 8.2.5 use a special nomenclature to define the read/write mode of individual bits in each register. See table bellow for details.

Table 8-5. PHY Register Nomenclature

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.

Note: For all binary equations appearing in the register map, the symbol "|" is equivalent to a binary OR operation.

8.1.2.1 Registers Byte Ordering

This section defines the structure of registers that contain fields carried over the network. Some examples are L2, L3, L4 fields and IPsec fields.

The following example is used to describe byte ordering over the wire (hex notation):

```
Last                        First
...,06, 05, 04, 03, 02, 01, 00
```

Each byte is sent with the LSbit first. That is, the bit order over the wire for this example is

```
Last                                       First
..., 0000 0011, 0000 0010, 0000 0001, 0000 0000
```

The general rule for register ordering is to use Host Ordering (also called little endian). Using the above example, a 6-byte fields (MAC address) is stored in a CSR in the following manner:

```
                   Byte 3  Byte 2  Byte 1  Byte0

DW address (N)     0x03    0x02    0x01    0x00
DW address (N+4)   ...     ...     0x05    0x04
```

The exceptions listed below use network ordering (also called big endian). Using the above example, a 16-bit field (EtherType) is stored in a CSR in the following manner:

```
                   Byte 3  Byte 2  Byte 1  Byte0
(DW aligned)       ...     ...     0x00    0x01
or
(Word aligned)     0x00    0x01    ...     ...
```

The following exceptions use network ordering:

All ETherType fields. For example, the VET EXT field in the VET register, the EType field in the ETQF register, the CMT_ETH in the RTTBCNCR register, the EType field in the METF register.

Note: The “normal” notation as it appears in text books, etc. is to use network ordering. Example: Suppose a MAC address of 00-A0-C9-00-00-00. The order on the network is 00, then A0, then C9, etc. However, the host ordering presentation would be:

```
                   Byte 3  Byte 2  Byte 1  Byte0
DW address (N)     00      C9      A0      00
DW address (N+4)   ...     ...     00      00
```

8.1.3 Register Summary

All the 82576's non-PCIe configuration registers, except for the MSI-X register, are listed in the table below. These registers are ordered by grouping and are not necessarily listed in order that they appear in the address space. In an IOV system, this list refers to the PF registers, the VF register space is listed in Section 8.25.

Table 8-6. Register Summary

Offset	Alias Offset	Abbreviation	Name	RW
		General		
0x0000	0x00004	CTRL	Device Control Register	RW
0x0008	N/A	STATUS	Device Status Register	RO
0x0018	N/A	CTRL_EXT	Extended Device Control Register	RW
0x0020	N/A	MDIC	MDI Control Register	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x0024	N/A	SERDESCTL	Serdes_ana	RW
0x0028	N/A	FCAL	Flow Control Address Low	RO
0x002C	N/A	FCAH	Flow Control Address High	RO
0x0030	N/A	FCT	Flow Control Type	RW
0x0034	N/A	CONNSW	Copper/Fiber switch control	RW
0x0038	N/A	VET	VLAN Ether Type	RW
0x0170	N/A	FCTTV	Flow Control Transmit Timer Value	RW
0x0E00	N/A	LEDCTL	LED Control Register	RW
0x1028	N/A	I2CCMD	SFP I2C command	RW
0x102C	N/A	I2CPARAMS	SFP I2C Parameter	RW
0x1040	N/A	WDSTP	Watchdog setup register	RW
0x1044	N/A	WDSWSTS	Watchdog Software	RW
0x1048	N/A	FRTIMER	Free Running Timer	RWS
0x104C	N/A	TCPTimer	TCP timer	RW
0x5B70	N/A	DCA_ID	DCA Requester ID Information Register	RO
0x05B50	N/A	SWSM	Software Semaphore Register	RW
0x05B54	N/A	FWSM	Firmware Semaphore Register	RWS
0x5B5C	N/A	SW_FW_SYNC	Software-Firmware Synchronization	RWS
Flash/EEPROM registers				
0x0010	N/A	EEC	EEPROM/Flash Control Register	RW
0x0014	N/A	EERD	EEPROM Read Register	RW
0x001C	N/A	FLA	Flash Access Register	RW
0x1010	N/A	EEMNGCTL	MNG EEPROM Control Register	RO
0x1014	N/A	EEMNGDATA	MNG EEPROM Read/Write data	RO
0x1018	N/A	FLMNGCTL	MNG Flash Control Register	RO
0x101C	N/A	FLMNGDATA	MNG Flash Read data	RO
0x1020	N/A	FLMNGCNT	MNG Flash Read Counter	RO
0x1024	N/A	EEARBC	EEPROM Auto Read Bus Control	RW
0x103C	N/A	FLASHOP	Flash Opcode Register	RW
0x1038	N/A	EEDIAG	EEPROM Diagnostic	RO
0x1060	N/A	VPDDIAG	VPD Diagnostic	RO
Interrupts				
0x01500	0x000C0	ICR	Interrupt Cause Read	RC/W1C
0x01504	0x000C8	ICS	Interrupt Cause Set	WO
0x01508	0x000D0	IMS	Interrupt Mask Set/Read	RW
0x0150C	0x000D8	IMC	Interrupt Mask Clear	WO
0x01510	0x000E0	IAM	Interrupt Acknowledge Auto Mask	RW
0x1520	N/A	EICS	Extended Interrupt Cause Set	WO
0x1524	N/A	EIMS	Extended Interrupt Mask Set/Read	RWS
0x1528	N/A	EIMC	Extended Interrupt Mask Clear	WO
0x152c	N/A	EIAC	Extended Interrupt Auto Clear	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x1530	N/A	EIAM	Extended Interrupt Auto Mask	RW
0x1580	N/A	EICR	Extended Interrupt Cause Read	RC/W1C
0x1700 - 0x171C	N/A	IVAR	Interrupt Vector Allocation Registers	RW
0x1740	N/A	IVAR_MISC	Interrupt Vector Allocation Registers - MISC	RW
0x1680 - 0x16F0	N/A	EITR	Extended Interrupt Throttling Rate 0 - 24	RW
0x1514	N/A	GPIE	General Purpose Interrupt Enable	RW
0x5B68	N/A	PBACL	MSI-X PBA Clear	R/W1C
Receive				
0x00100	N/A	RCTL	RX Control	RW
0x02160	0x00168	FCRTL0	Flow Control Receive Threshold Low	RW
0x02170	N/A	FCRTL1	Flow Control Receive Threshold Low	RW
0x02458	N/A	PBDIAG	PB Diagnostic	RW
0x02404	N/A	RXPBS	RX packet buffer size	RW
0x24E8	N/A	PBRWAC	Rx Packet Buffer wrap around counter	RO
0x02460	N/A	FCRTV	Flow control Refresh timer value	RW
0x2540	N/A	DRXMXOD	DMA RX Max Total Allow Size Requests	RW
0xC000	0x00110, 0x02800	RDBAL[0]	RX Descriptor Base Low queue 0	RW
0xC004	0x00114, 0x02804	RDBAH[0]	RX Descriptor Base High queue 0	RW
0xC008	0x00118, 0x02808	RDLEN[0]	RX Descriptor Ring Length queue 0	RW
0xC00C	0x280C	SRRCTL[0]	Split and Replication Receive Control Register queue 0	RW
0xC010	0x00120, 0x02810	RDH[0]	RX Descriptor Head queue 0	RW
0xC018	0x00128, 0x2818	RDT[0]	RX Descriptor Tail queue 0	RW
0xC028	0x02828	RXDCTL[0]	Receive Descriptor Control queue 0	RW
0xC014	0x2814	RXCTL[0]	Rx Queue 0 DCA CTRL Register	RW
0xC030	0x2830	RQDPC[0]	Rx Queue drop packet count Register 0	RC
0xC040 + 0x40 * (n-1)	0x2900+ 0x100 * (n-1)	RDBAL[1 - 3]	RX Descriptor Base Low queue 1 - 3	RW
0xC044 + 0x40 * (n-1)	0x2904 + 0x100 * (n-1)	RDBAH[1 - 3]	RX Descriptor Base High queue 1 - 3	RW
0xC048 + 0x40 * (n-1)	0x2908 + 0x100 * (n-1)	RDLEN[1 - 3]	RX Descriptor Ring Length queue 1 - 3	RW
0xC04C + 0x40 * (n-1)	0x290C + 0x100 * (n-1)	SRRCTL[1 - 3]	Split and Replication Receive Control Register queue 1 - 3	RW
0xC050 + 0x40 * (n-1)	0x2910 + 0x100 * (n-1)	RDH[1 - 3]	RX Descriptor Head queue 1 - 3	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0xC058 + 0x40 * (n-1)	0x2918 + 0x100 * (n-1)	RDT[1 - 3]	RX Descriptor Tail queue 1 - 3	RW
0xC068 + 0x40 * (n-1)	0x2928 + 0x100 * (n-1)	RXDCTL[1 - 3]	Receive Descriptor Control queue 1 - 3	RW
0xC054 + 0x40 * (n-1)	0x2914 + 0x100 * (n-1)	RXCTL[1 - 3]	Rx Queue 1 - 3 DCA CTRL Register	RW
0xC070 + 0x40 * (n-1)	0x2930 + 0x100 * (n-1)	RQDPC[1 - 3]	Rx Queue drop packet count Register 1 - 3	RC
0xC100 + 0x40 * (n- 4)	N/A	RDBAL[4-15]	RX Descriptor Base Low queue 4 - 15	RW
0xC104 + 0x40 * (n- 4)	N/A	RDBAH[4-15]	RX Descriptor Base High queue 4 - 15	RW
0xC108 + 0x40 * (n- 4)	N/A	RDLEN[4-15]	RX Descriptor Ring Length queue 4 - 15	RW
0xC10C + 0x40 * (n- 4)	N/A	SRRCTL[4 -15]	Split and Replication Receive Control Register queue 4 - 15	RW
0xC110 + 0x40 * (n- 4)	N/A	RDH[4 - 15]	RX Descriptor Head queue 4 - 15	RW
0xC118 + 0x40 * (n- 4)	N/A	RDT[4 - 15]	RX Descriptor Tail queue 4 - 15	RW
0xC128 + 0x40 * (n- 4)	N/A	RXDCTL[4 - 15]	Receive Descriptor Control queue 4 - 15	RW
0xC114 + 0x40 * (n- 4)	N/A	RXCTL[4 - 15]	Rx Queue 4 - 15 DCA CTRL Register	RW
0xC130 + 0x40 * (n- 4)	N/A	RQDPC[4 - 15]	Rx Queue drop packet count Register 8 -15	RC
0x05000	N/A	RXCSUM	Receive Checksum Control	RW
0x05004	N/A	RLPML	Receive Long packet maximal length	RW
0x05008	N/A	RFCTL	Receive Filter Control Register	RW
0x05200-0x053FC	0x00200-0x003FC	MTA[127:0]	Multicast Table Array (n)	RW
0x05400 + 8*n	0x00040 + 8*n	RAL[0-15]	Receive Address Low (15:0)	RW
0x05404 + 8 *n	0x00044 + 8 *n	RAH[0-15]	Receive Address High (15:0)	RW
0x054E0 + 8*n	N/A	RAL[16-23]	Receive Address Low (23:16)	RW
0x054E4 + 8 *n	N/A	RAH[16-23]	Receive Address High (23:16)	RW
0x5480 – 0x549C	N/A	PSRTYPE[7:0]	Packet Split Receive type (n)	RW
0x54C0	N/A	RPLPSRTYPE	Replicated Packet Split Receive type	RW
0x581C	N/A	VT_CTL	Next Generation VMDq Control register	RW
0x05600-0x057FC	0x00600-0x007FC	VFTA[127:0]	VLAN Filter Table Array (n)	RW
0x05818	N/A	MRQC	Multiple Receive Queues Command	RW
0x05C00-0x05C7C	N/A	RETA	Redirection Table	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x05C80-0x05CA4	N/A	RSSRK	RSS Random Key Register	RW
0x34E8	N/A	PBTWAC	Tx Packet Buffer wrap around counter	RO
Transmit				
0x03404	N/A	TXPBS	TX packet buffer size	RW
0x00400	N/A	TCTL	TX Control	RW
0x00404	N/A	TCTL_EXT	TX Control extended	RW
0x00410	N/A	TIPG	TX IPG	RW
0x3590	N/A	DTXCTL	DMA TX Control	RW
0x359C	N/A	DTXTCPFLGL	DMA TX TCP Flags Control Low	RW
0x35A0	N/A	DTXTCPFLGH	DMA TX TCP Flags Control High	RW
0x3540	N/A	DTXMXSZRQ	DMA TX Max Total Allow Size Requests	RW
0xE000	0x00420, 0x03800	TDBAL[0]	TX Descriptor Base Low 0	RW
0xE004	0x00424, 0x03804	TDBAH[0]	TX Descriptor Base High 0	RW
0xE008	0x00428, 0x03808	TDLEN[0]	TX Descriptor Ring Length 0	RW
0xE010	0x00430, 0x03810	TDH[0]	TX Descriptor Head 0	RW
0xE018	0x00438, 0x03818	TDT[0]	TX Descriptor Tail 0	RW
0xE028	0x03828	TXDCTL[0]	Transmit Descriptor Control queue 0	RW
0xE014	0x3814	TXCTL[0]	TX DCA CTRL Register Queue 0	RW
0xE038	0x3838	TDWBAL[0]	Transmit Descriptor WB Address Low queue 0	RW
0xE03C	0x383C	TDWBAH[0]	Transmit Descriptor WB Address High queue 0	RW
0x0E040 + 0x100 * (n-1)	0x03900 + 0x100 * (n-1)	TDBAL[1-3]	TX Descriptor Base Low queue 1 - 3	RW
0x0E044 + 0x100 * (n-1)	0x03904 + 0x100 * (n-1)	TDBAH[1-3]	TX Descriptor Base High queue 1 - 3	RW
0x0E048 + 0x100 * (n-1)	0x03908 + 0x100 * (n-1)	TDLEN[1-3]	TX Descriptor Ring Length queue 1 - 3	RW
0x0E050 + 0x100 * (n-1)	0x03910 + 0x100 * (n-1)	TDH[1-3]	TX Descriptor Head queue 1 - 3	RW
0x0E058 + 0x100 * (n-1)	0x03918 + 0x100 * (n-1)	TDT[1-3]	TX Descriptor Tail queue 1 - 3	RW
0x0E068 + 0x100 * (n-1)	0x03928 + 0x100 * (n-1)	TXDCTL[1-3]	Transmit Descriptor Control 1 - 3	RW
0x0E054 + 0x100 * (n-1)	0x3914 + 0x100 * (n-1)	TXCTL[1-3]	TX DCA CTRL Register Queue 1 - 3	RW
0x0E078 + 0x100 * (n-1)	0x3938 + 0x100 * (n-1)	TDWBAL[1-3]	Transmit Descriptor WB Address Low queue 1 - 3	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x0E07C + 0x100 * (n-1)	0x393C + 0x100 * (n-1)	TDWBAH[1-3]	Transmit Descriptor WB Address High queue 1 - 3	RW
0x0E180 + 0x100 *n	N/A	TDBAL[4 - 15]	TX Descriptor Base Low queue 4 - 15	RW
0x0E184 + 0x100 * n	N/A	TDBAH[4 - 15]	TX Descriptor Base High queue 4 - 15	RW
0x0E188 + 0x100 * n	N/A	TDLEN[4 - 15]	TX Descriptor Ring Length queue 4 - 15	RW
0x0E190 + 0x100 * n	N/A	TDH[4 - 15]	TX Descriptor Head queue 4 - 15	RW
0x0E198 + 0x100 * n	N/A	TDT[4 - 15]	TX Descriptor Tail queue 4 - 15	RW
0x0E1A8 + 0x100 * n	N/A	TXDCTL[4 - 15]	Transmit Descriptor Control 4 - 15	RW
0x0E194 + 0x100 * n	N/A	TXCTL[4 - 15]	TX Queue 4 - 15 DCA CTRL Register	RW
0x0E1B8 + 0x100 * n	N/A	TDWBAL[4 - 15]	Transmit Descriptor WB Address Low queue 4 - 15	RW
0x0E1BC + 0x100 * n	N/A	TDWBAH[4 - 15]	Transmit Descriptor WB Address High queue 4 - 15	RW
Filters				
0x5CB0 - 0x5CCC	N/A	ETQF[0 - 7]	EType Queue Filter 0 - 7	RW
0x05A80 - 0x05A9C	N/A	IMIR[0 - 7]	Immediate Interrupt Rx [7:0]	RW
0x5AA0 - 0x5ABC	N/A	IMIREXT[0 - 7]	Immediate Interrupt Rx Extended[0-7]	RW
0x05AC0	N/A	IMIRVP	Immediate Interrupt Rx VLAN Priority	RW
0x5980 - 0x599C	N/A	SAQF[0 - 7]	Source Address Queue Filter 0 - 7	RW
0x59A0 - 0x59BC	N/A	DAQF[0 - 7]	Destination Address Queue Filter 0 - 7	RW
0x59C0 - 0x59DC	N/A	SPQF[0 - 7]	Source Port Queue Filter 0 - 7	RW
0x59E0 - 0x59FC	N/A	FTQF[0 - 7]	Five-Tuple Queue Filter 0 - 7	RW
0x55FC	N/A	SYNQF	SYN Packet Queue Filter	RW
Virtualization				
0x03004	N/A	SWPBS	Switch packet buffer size	RW
0x030E8	N/A	PBSWAC	Switch Packet Buffer wrap around counter	RO
0x00C40 - 0x00C5C	N/A	VFMailbox[0 - 7]	VF mailbox register	RW
0x00C00 - 0x00C1C	N/A	PFMailbox[0 - 7]	PF Mailbox register	RW
0x00800 - 0x009FC	N/A	VMBMEM	Virtual machines Mailbox Memory	RW
0x0C80	N/A	MBVFICR	Mailbox VF interrupt causes	R/W1C
0x0C84	N/A	MBVFIMR	Mailbox VF interrupt mask	RW
0x0C88	N/A	VFLRE	VFLR Events	R/W1C

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x0C8C	N/A	VFRE	VF Receive Enable	RW
0x0C90	N/A	VFTE	VF Transmit Enable	RW
0x3554	N/A	WVBR	Wrong VM Behavior Register	RC/W1C
0x3510	N/A	VMECM	VM Error count mask	RW
0x3548	N/A	LVMMC	Last VM Misbehavior Cause	RC
0x3558	N/A	MDFB	Malicious Driver Free Block	RWS
0x3550	N/A	DTXMAPT	DMA max arbitration time	RW
0x02408	N/A	QDE	Queue drop enable register	RW
0x3500	N/A	DTXSWC	DMA Tx Switch control	RW
0x05D00 - 0x05D7C	N/A	VLVF	VLAN VM Filter	RW
0x05AD0 - 0x5AEC	N/A	VMOLR[0 - 7]	VM Offload register[0-7]	RW
0x03700	N/A	VMVIR	VM VLAN insert register	RW
0x05AF0	N/A	RPLOLR	Replication Offload register	RW
0x0A000 - 0x0A1FC	N/A	UTA	Unicast Table Array	WO
0x5D80 - 0x5D8C	N/A	VMRCTL	Virtual Mirror rule control	RW
0x5D90 - 0x5D9C	N/A	VMRVLAN	Virtual Mirror rule VLAN	RW
0x5DA0 - 0x5DAC	N/A	VMRVM	Virtual Mirror rule VM	RW
0x5DB0	N/A	SCCRL	Storm Control control register	RW
0x5DB4	N/A	SCSTS	Storm Control status	RO
0x5DB8	N/A	BSCTRH	Broadcast Storm control Threshold	RW
0x5DBC	N/A	MSCTRH	Multicast Storm control Threshold	RW
0x5DC0	N/A	BSCCNT	Broadcast Storm Control Current Count	RO
0x5DC4	N/A	MSCCNT	Multicast Storm control Current Count	RO
0x5DC8	N/A	SCTC	Storm Control Time Counter	RO
Statistics				
0x04000	N/A	CRCERRS	CRC Error Count	RC
0x04004	N/A	ALGNERRC	Alignment Error Count	RC
0x04008	N/A	SYMERRS	Symbol Error Count	RC
0x0400C	N/A	RXERRC	RX Error Count	RC
0x04010	N/A	MPC	Missed Packets Count	RC
0x04014	N/A	SCC	Single Collision Count	RC
0x04018	N/A	ECOL	Excessive Collisions Count	RC
0x0401C	N/A	MCC	Multiple Collision Count	RC
0x04020	N/A	LATECOL	Late Collisions Count	RC
0x04028	N/A	COLC	Collision Count	RC
0x04030	N/A	DC	Defer Count	RC
0x04034	N/A	TNCRS	Transmit - No CRS	RC

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x0403C	N/A	HTDPMC	Host Transmit Discarded Packets by MAC Count	RC
0x04040	N/A	RLEC	Receive Length Error Count	RC
0x04048	N/A	XONRXC	XON Received Count	RC
0x0404C	N/A	XONTXC	XON Transmitted Count	RC
0x04050	N/A	XOFFRXC	XOFF Received Count	RC
0x04054	N/A	XOFFTXC	XOFF Transmitted Count	RC
0x04058	N/A	FCRUC	FC Received Unsupported Count	RC
0x0405C	N/A	PRC64	Packets Received (64 Bytes) Count	RC
0x04060	N/A	PRC127	Packets Received (65-127 Bytes) Count	RC
0x04064	N/A	PRC255	Packets Received (128-255 Bytes) Count	RC
0x04068	N/A	PRC511	Packets Received (256-511 Bytes) Count	RC
0x0406C	N/A	PRC1023	Packets Received (512-1023 Bytes) Count	RC
0x04070	N/A	PRC1522	Packets Received (1024-1522 Bytes)	RC
0x04074	N/A	GPRC	Good Packets Received Count	RC
0x04078	N/A	BPRC	Broadcast Packets Received Count	RC
0x0407C	N/A	MPRC	Multicast Packets Received Count	RC
0x04080	N/A	GPTC	Good Packets Transmitted Count	RC
0x04088	N/A	GORCL	Good Octets Received Count (Lo)	RC
0x0408C	N/A	GORCH	Good Octets Received Count (Hi)	RC
0x04090	N/A	GOTCL	Good Octets Transmitted Count (Lo)	RC
0x04094	N/A	GOTCH	Good Octets Transmitted Count (Hi)	RC
0x040A0	N/A	RNBC	Receive No Buffers Count	RC
0x024E0 0x024E4	0x040A0	TCRNBC[1:0]	TC Receive No Buffers Count [1:0]	RC
0x040A4	N/A	RUC	Receive Under size Count	RC
0x040A8	N/A	RFC	Receive Fragment Count	RC
0x040AC	N/A	ROC	Receive Oversize Count	RC
0x040B0	N/A	RJC	Receive Jabber Count	RC
0x040B4	N/A	MNGPRC	Management Packets Receive Count	RC
0x040B8	N/A	MPDC	Management Packets Dropped Count	RC
0x040BC	N/A	MNGPTC	Management Packets Transmitted Count	RC
0x413c	N/A	BMNGPRC	BMC Management Packets Receive Count	RC
0x4140	N/A	BMPDC	BMC Management Packets Dropped Count	RC
0x4144	N/A	BMNGPTC	BMC Management Packets Transmitted Count	RC
0x040C0	N/A	TORL	Total Octets Received (Lo)	RC
0x040C4	N/A	TORH	Total Octets Received (Hi)	RC
0x040C8	N/A	TOTL	Total Octets Transmitted (Lo)	RC
0x040CC	N/A	TOTH	Total Octets Transmitted (Hi)	RC
0x040D0	N/A	TPR	Total Packets Received	RC
0x040D4	N/A	TPT	Total Packets transmitted	RC
0x040D8	N/A	PTC64	Packets Transmitted (64 Bytes) Count	RC
0x040DC	N/A	PTC127	Packets Transmitted (65-127 Bytes) Count	RC

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x040E0	N/A	PTC255	Packets Transmitted (128-256 Bytes) Count	RC
0x040E4	N/A	PTC511	Packets Transmitted (256-511 Bytes) Count	RC
0x040E8	N/A	PTC1023	Packets Transmitted (512-1023 Bytes) Count	RC
0x040EC	N/A	PTC1522	Packets Transmitted (1024-1522 Bytes) Count	RC
0x040F0	N/A	MPTC	Multicast Packets Transmitted Count	RC
0x040F4	N/A	BPTC	Broadcast Packets Transmitted Count	RC
0x040F8	N/A	TSCTC	TCP Segmentation Context Transmitted Count	RC
0x04100	N/A	IAC	Interrupt Assertion Count	RC
0x04104	N/A	RPTHC	Rx Packets to host count	RC
0x04108	N/A	DBGC1	Debug counter 1	RC
0x0410C	N/A	DBGC2	Debug counter 2	RC
0x04110	N/A	DBGC3	Debug counter 3	RC
0x0411C	N/A	DBGC4	Debug counter 4	RC
0x04118	N/A	HGPTC	Host Good Packets Transmitted Count	RC
0x04120	N/A	RXDMTC	Rx Descriptor Minimum Threshold Count	RC
0x04128	N/A	HGORCL	Host Good Octets Received Count (Lo)	RC
0x0412C	N/A	HGORCH	Host Good Octets Received Count (Hi)	RC
0x04130	N/A	HGOTCL	Host Good Octets Transmitted Count (Lo)	RC
0x04134	N/A	HGOTCH	Host Good Octets Transmitted Count (Hi)	RC
0x4138	N/A	LENERRS	Length Errors count register	RC
0x4228	N/A	SCVPC	SerDes/SGMII Code Violation Packet Count Register	RW
0x41A0	N/A	SSVPC	Switch Security Violation Packet Count	RC
0x41A4	N/A	SDPC	Switch Drop Packet Count	RC
0x4300- 0x43D0	N/A	Reserved	Reserved	
Wake up				
0x05800	N/A	WUC	Wake Up Control	RW
0x05808	N/A	WUFC	Wake Up Filter Control	RW
0x05810	N/A	WUS	Wake Up Status	R/W1C
0x05900	N/A	WUPL	Wake Up Packet Length	RO
0x05A00- 0x05A7C	N/A	WUPM	Wake Up Packet Memory	RO
0x09000- 0x093FC	N/A	FHFT	Flexible Host Filter Table registers	RW
0x09A00- 0x09BFC	N/A	FHFT_EXT	Flexible Host Filter Table registers extended	RW
Manageability				
0x05010 - 0x0502C	N/A	MAVTV[7:0]	VLAN TAG Value 7 - 0	RW
0x5030 - 0x504C	N/A	MFUTP[7:0]	Management Flex UDP/TCP Ports	RW
0x05060 - 0x0506C	N/A	METF[3:0]	Management Ethernet Type Filters	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x05820	N/A	MANC	Management Control	RW
0x05838	N/A	IPAV	IP Address Valid	RW
0x5824	N/A	MFVAL	Manageability Filters Valid	RW
0x05840-0x05858	N/A	IP4AT	IPv4 Address Table	RW
0x05860	N/A	MANC2H	Management Control To Host Register	RW
0x05880-0x0588F	N/A	IP6AT	IPv6 Address Table	RW
0x5890 - 0x58AC	N/A	MDEF[7:0]	Manageability Decision Filters	RW
0x5930 – 0x594C	N/A	MDEF_EXT[7:0]	Manageability Decision Filters	RW
0x58B0 - 0x58EC	N/A	MIPAF	Manageability IP Address Filter	RW
0x5910 + 8*n	N/A	MMAL[3:0]	Manageability MAC Address Low 3:0	RW
0x5914 + 8*n	N/A	MMAH[3:0]	Manageability MAC Address High 3:0	RW
0x09400-0x097FC	N/A	FTFT	Flexible TCO Filter Table	RW
0x08800-0x08EFC	N/A	Flex MNG	Flex manageability memory address space	RW
0x08F14	N/A	Reserved	Reserved	
PCIe				
0x05B00	N/A	GCR	PCIe Control Register	RW
0x05B04	N/A	RTIV	Replay Timer Initial Value Register	RW
0x05B08	N/A	FUNCTAG	Function Tag register	RW
0x05B88	N/A	CIAA	Config Indirect Access Address	RW
0x05B8C	N/A	CIAD	Config Indirect Access Data	RW
0x5BBC	N/A	IOVCTL	IOV control	RW
0x05B0C	N/A	LTIV	Latency Timer Initial Value Register	RW
0x05B10	N/A	GSCL_1	PCIe statistics control #1	RW
0x05B14	N/A	GSCL_2	PCIe statistics control #2	RW
0x05B18	N/A	GSCL_3	PCIe statistics control #3	RW
0x05B1C	N/A	GSCL_4	PCIe statistics control #4	RW
0x5B90 - 0x5B9C	N/A	GSCL_LBT[3:0]	PCIe statistics control Leaky Bucket Timer	RW
0x05B20	N/A	GSCN_0	PCIe counter register #0	RW
0x05B24	N/A	GSCN_1	PCIe counter register #1	RW
0x05B28	N/A	GSCN_2	PCIe counter register #2	RW
0x05B2C	N/A	GSCN_3	PCIe counter register #3	RW
0x05B30	N/A	FACTPS	Function Active and Power State	RW
0x05B34	N/A	GIOANACTL0	SerDes/CCM/PCIe CSR	RW
0x05B38	N/A	GIOANACTL1	SerDes/CCM/PCIe CSR	RW
0x05B3C	N/A	GIOANACTL2	SerDes/CCM/PCIe CSR	RW
0x05B40	N/A	GIOANACTL3	SerDes/CCM/PCIe CSR	RW

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x05B44	N/A	GIOANACTLALL	SerDes/CCM/PCIe CSR	RW
0x05B48	N/A	CCMCTL	SerDes/CCM/PCIe CSR	RW
0x05B4C	N/A	SCCTL	SerDes/CCM/PCIe CSR	RW
0x05B64	N/A	MREVID	Mirrored Revision ID	RO
0x05B74	N/A	DCA_CTRL	DCA Control Register	RW
Diagnostic				
0x02410	0x08000	RDFH	RX Data FIFO Head	RWS
0x02418	0x08008	RDFT	RX Data FIFO Tail	RWS
0x02420	N/A	RDFHS	RX Data FIFO Head Saved	RWS
0x02428	N/A	RDFTS	RX Data FIFO Tail Saved	RWS
0x02430	N/A	RDFPC	Receive Data FIFO Packet Count	RO
0x3010	N/A	SWBFH	Switch Buffer FIFO Head	RWS
0x3018	N/A	SWBFT	Switch Buffer FIFO Tail	RWS
0x3020	N/A	SWBFHS	Switch Buffer FIFO Head Saved	RWS
0x3028	N/A	SWBFTS	Switch Buffer FIFO Tail Saved	RWS
0x03030	N/A	SWDFPC	Switch Data FIFO Packet Count	RO
0x245C	N/A	RPBECCSTS	Receive Packet buffer ECC control	RC
0x345C	N/A	TPBECCSTS	Transmit Packet buffer ECC control	RC
0x305C	N/A	SWPBECCSTS	Switch Packet buffer ECC control	RC
0xB470	N/A	IPPBECCSTS	IPsec Packet buffer ECC control	RC
0x2464	N/A	FCSTS0	Flow Control Status	RO
0x25C0	N/A	RDHESTS	Rx Descriptor Handler ECC status	RC
0x35C0	N/A	TDHESTS	Tx Descriptor Handler ECC status	RC
0x05BB0	N/A	PWBESTS	PCIe Write Buffer ECC Status	RC
0x05BA8	N/A	PMSIXESTS	PCIe MSI-X ECC Status	RC
0x05BA0	N/A	PRBESTS	PCIe Retry Buffer ECC Status	RC
0x0B474	N/A	IPPBEEI	IPSec Packet Buffer ECC Error inject	RW
0x25FC	N/A	RDHMP	Rx Descriptor Handler Memory Page Number	RW
0x03410	0x08010	TDFH	TX Data FIFO Head	RWS
0x03418	0x08018	TDFT	TX Data FIFO Tail	RWS
0x03420	N/A	TDFHS	TX Data FIFO Head Saved	RWS
0x03428	N/A	TDFTS	TX Data FIFO Tail Saved	RWS
0x03430	N/A	TDFPC	Transmit Data FIFO Packet Count	RO
0x35FC	N/A	TDHMP	Tx Descriptor Handler Memory Page Number	RW
0x6000 - 0x6FFC	N/A	RDHM	RX descriptors internal cache	RO
0x7000 - 0x7FFC	N/A	TDHM	TX descriptors internal cache	RO
0x3100	N/A	PBSLAC	PB Slave Access Control	RW
0x3110 - 0x311C	N/A	PBSLAD	PB Slave Access Data	RW
0x1084	N/A	PEIND	Parity and ECC Indication	RC

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x1088	N/A	PEINDM	Parity and ECC Indication Mask	RW
0x5BB8	N/A	FUNCTO	Function Timeout	RO
0x00F00	N/A	CIRC	Circuits Control	RO
0x35E0	N/A	TXBDC	Tx DMA Performance Burst and Descriptor Count	RC
0x35E4	N/A	TXIDLE	Tx DMA Performance Idle Count	RC
0x25E0	N/A	RXBDC	Tx DMA Performance Burst and Descriptor Count	RC
0x25E4	N/A	RXIDLE	Tx DMA Performance Idle Count	RC
0x05B7C	N/A	ULT1	ULT1 Register	RO
0x05B80	N/A	ULT2	ULT2 Register	RO
0x05B84	N/A	Strap	Strap Register	RO
PCS				
0x4200	N/A	PCS_CFG	PCS Configuration 0 Register	RW
0x4208	N/A	PCS_LCTL	PCS Link Control Register	RW
0x420C	N/A	PCS_LSTS	PCS Link Status Register	RO
0x4210	N/A	PCS_DBG0	PCS Debug 0 register	RO
0x4214	N/A	PCS_DBG1	PCS Debug 1 register	RO
0x4218	N/A	PCS_ANADV	AN advertisement Register	RW
0x421C	N/A	PCS_LPAB	Link Partner Ability Register	RO
0x4220	N/A	PCS_NPTX	AN Next Page Transmit Register	RW
0x4224	N/A	PCS_LPABNP	Link Partner Ability Next Page Register	RO
Time Sync				
0x0B620	N/A	TSYNCRXCTL	RX Time Sync Control register	RW
0x0B624	N/A	RXSTMPL	RX timestamp Low	RO
0x0B628	N/A	RXSTMPH	RX timestamp High	RO
0x0B62C	N/A	RXSATRL	RX timestamp attributes low	RO
0x0B630	N/A	RXSATRH	RX timestamp attributes low	RO
0x0B614	N/A	TSYNCTXCTL	TX Time Sync Control register	RW
0x0B618	N/A	TXSTMPL	TX timestamp value Low	RO
0x0B61C	N/A	TXSTMPH	TX timestamp value High	RO
0x0B600	N/A	SYSTIML	System time register Low	RWS
0x0B604	N/A	SYSTIMH	System time register High	RWS
0x0B608	N/A	TIMINCA	Increment attributes register	RW
0x0B60C	N/A	TIMADJL	Time adjustment offset register low	RW
0x0B610	N/A	TIMADJH	Time adjustment offset register high	RW
0x0B640	N/A	TSAUXC	Auxiliary Control Register	RW
0x0B644	N/A	TRGTTIML0	Target Time register 0 Low	RW
0x0B648	N/A	TRGTTIMH0	Target Time register 0 High	RW
0x0B64C	N/A	TRGTTIML1	Target Time register 1 Low	RW
0x0B650	N/A	TRGTTIMH1	Target Time register 1 High	RW
0x0B65C	N/A	AUXSTMPL0	Auxiliary Time Stamp 0 register Low	RO
0x0B660	N/A	AUXSTMPH0	Auxiliary Time Stamp 0 register High	RO

Table 8-6. Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x0B664	N/A	AUXSTMPL1	Auxiliary Time Stamp 1 register Low	RO
0x0B668	N/A	AUXSTMPH1	Auxiliary Time Stamp 1 register High	RO
0x05F50	N/A	TSYNCRXCFG	Time Sync RX Configuration	RW
0x0003C	N/A	TSSDP	Time Sync SDP Config Reg	RW
IPSec				
0xB408	N/A	IPSRXCMD	IPsec Rx Command Register	RW
0xB400	N/A	IPSRXIDX	IPsec Rx Index	RW
0xB420 - 0xB42C	N/A	IPSRXIPADDR	IPsec Rx IP address Register	RW
0xB410 - 0xB41C	N/A	IPSRXKEY	IPsec Rx Key Register	RW
0xB404	N/A	IPSRXSALT	IPsec Rx Salt Register	RW
0xB40C	N/A	IPSRXSPI	IPsec Rx SPI Register	RW
0xB450	N/A	IPSTXIDX	IPsec Tx Index	RW
0xB460 - 0xB46C	N/A	IPSTXKEY	IPsec Tx Key Registers	RW
0xB454	N/A	IPSTXSALT	IPsec Tx Salt Register	RW
0xB430	N/A	IPSCTRL	IPSec Control	RW

Certain registers maintain an alias address designed for backward compatibility with software written for previous GbE controllers. For these registers, the alias address is shown in the table above. Those registers can be accessed by software at either the new offset or the alias offset. It is recommended that software that is written solely for the 82576, use the new address offset.

8.1.4 MSI-X BAR Register Summary

Table 8-7. MSI-X Register Summary

Category	Offset	Abbreviation	Name	RW	Page
MSI-X Table	0x0000 + n*0x10 [n=0...24]	MSIXTADD	MSI–X Table Entry Lower Address	RW	page 421
MSI-X Table	0x0004 + n*0x10 [n=0..24]	MSIXTUADD	MSI–X Table Entry Upper Address	RW	page 421
MSI-X Table	0x0008 + n*0x10 [n=0...24]	MSIXTMSG	MSI–X Table Entry Message	R/W	page 421
MSI-X Table	0x000C + n*0x10 [n=0...24]	MSIXTVCTRL	MSI–X Table Entry Vector Control	R/W	page 421
MSI-X Table	0x02000	MSIXPBA	MSIXPBA Bit Description	RO	page 422

8.2 General Register Descriptions

8.2.1 Device Control Register - CTRL (0x00000; R/W)

This register, as well as the Extended Device Control register (CTRL_EXT), controls the major operational modes for the device. While software write 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. See Section 4.5.7 for details on the setup of these registers in the different link modes.

Note: This register is also aliased at address 0x0004.

Field	Bit(s)	Initial Value	Description
FD	0	1b[1]	Full-Duplex Controls the MAC duplex setting when explicitly set by software. 0b = half duplex. 1b = full duplex.
Reserved	1	0b	This bit is reserved and should be set to 0b for future compatibility.
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.
Link Reset	3	1b[1]	Link Reset. 0 = normal; 1 = reset. Used to reset/restart the link auto-negotiation process when using SerDes mode.
Reserved	4	0b	Reserved
Reserved	5	0b[1]	Reserved Must be set to 0b. - Was ASDE
SLU	6	0b[1]	Set Link Up. When the MAC link mode is set for GMII/MII mode (internal PHY), Set Link Up must be set to 1 to permit the MAC to recognize the LINK signal from the PHY, which indicates the PHY has gotten the link up, and is ready to receive and transmit data. See Section 3.5.4 for more information about Auto-Negotiation and link configuration in the various modes. The “Set Link Up” is normally initialized to 0. However, if the APM Enable bit is set in the EEPROM then it is initialized to 1b.
ILOS	7	0b[1]	Invert Loss-of-Signal (LOS/LINK) Signal 0b = Do not invert (active high input signal). 1b = Invert signal (active low input signal). Should be set to zero when using internal PHY.
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. These signals are ignored when Auto-Speed Detection is enabled. 00b = 10 Mb/s. 01b = 100 Mb/s. 10b = 1000 Mb/s. 11b = not used.
Reserved	10	0b	Reserved Write as 0b to ensure future compatibility.

Field	Bit(s)	Initial Value	Description
FRCSPD	11	0b[1]	Force Speed This bit is set when software needs to manually configure the MAC speed settings according to the SPEED bits. When using a PHY, note that it must resolve to the same speed configuration or software must manually set it to the same speed as the MAC. The default is asserted. Software must clear this bit to enable the PHY or ASD function to control the MAC speed setting. Note that this bit is superseded by the CTRL_EXT.SPD_BYPS bit which has a similar function.
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 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	15:13	0b	Reserved Reads as 0b.
SDP0_GPIEN	16	0b	General Purpose Interrupt Detection Enable for SDP0 If software-controlled IO pin SDP0 is configured as an input, this bit (when 1b) enables the use for GPI interrupt detection.
SDP1_GPIEN	17	0b	General Purpose Interrupt Detection Enable for SDP1 If software-controlled IO pin SDP1 is configured as an input, this bit (when 1b) enables the use for GPI interrupt detection.
SDP0 DATA (RWS)	18	0b[1]	SDP0 Data Value Used to read or write the value of software-controlled 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. When the SDP0_WDE bit is set, this field indicates the polarity of the watchdog indication.
SDP1 DATA (RWS)	19	0b[1]	SDP1 Data Value Used to read or write the value of software-controlled IO pin SDP1. If SDP1 is configured as an output (SDP1_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP1 is configured as an input, reads return the current value of the pin.
ADVD3WUC	20	1b[1]	D3Cold Wake up Capability Advertisement Enable When set, D3Cold wake up capability is advertised based on whether AUX_PWR advertises presence of auxiliary power (yes if AUX_PWR is indicated, no otherwise). When 0b, however, D3Cold wake up capability is not advertised even if AUX_PWR presence is indicated. Note that the initial value is EEPROM configurable. If full 1Gb/sec. operation in D3 state is desired but the system's power requirements in this mode would exceed the D3Cold Wake up-Enabled specification limit (375mA at 3.3V), this bit can be used to prevent the capability from being advertised to the system.
SDP0_WDE	21	0b[1]	SDP0 used for Watchdog indication When set, SDP0 is used as a watchdog indication. When set, the SDP0_DATA bit indicates the polarity of the watchdog indication. In this mode, SDP0_IODIR must be set to an output.
SDP0_IODIR	22	0b[1]	SDP0 Pin Direction 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.

Field	Bit(s)	Initial Value	Description
SDP1_IODIR	23	0b[1]	SDP1 Pin Direction Controls whether software-controllable pin SDP1 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
Reserved	25:24	0b[1]	Reserved. Formerly used as SDP3 and SDP2 pin input/output direction control, respectively.
RST	26	0b	Device Reset This bit performs a reset of the entire controller device, resulting in a state nearly approximating the state following a power-up reset or internal PCIe reset, except for system PCI configuration. 0b = Normal. 1b = Reset. This bit is self clearing and is referred to as software reset or global reset.
RFCE	27	0b	Receive Flow Control Enable When set, indicates that the the 82576 responds to the reception of flow control packets. If Auto-Negotiation is enabled, this bit should be set to the negotiated flow control value. Is SerDes mode the resolution is done by the hardware. In internal PHY or SGMII modes it should be done by the software.
TFCE	28	0b	Transmit Flow Control Enable When set, indicates that the the 82576 transmits flow control packets (XON and XOFF frames) based on the receiver fullness. If Auto-Negotiation is enabled, this bit should be set to the negotiated flow control value. Is SerDes mode the resolution is done by the hardware. In internal PHY or SGMII modes it should be done by the software.
Reserved	29	0b	Reserved
VME	30	0b	VLAN Mode Enable When set to 1b, VLAN information is stripped from all received 802.1Q packets.
PHY_RST	31	0b	PHY Reset Controls a hardware-level reset to the internal PHY. 0b = Normal operation. 1b = PHY reset asserted.

[1] If the signature bits of the EEPROM's Initialization Control Word 1 match (01b), these bits are read from the EEPROM.

8.2.2 Device Status Register - STATUS (0x00008; R)

Field	Bit(s)	Initial Value	Description
FD	0	X	Full Duplex. 0 = half duplex; 1= full duplex. Reflects duplex setting of the MAC and/or link. FD reflects the actual MAC duplex configuration. This normally reflects the duplex setting for the entire link, as it normally reflects the duplex configuration negotiated between the PHY and link partner (copper link) or MAC and link partner (fiber link).
LU	1	X	Link up. 0 = no link established; 1 = link established. For this bit to be valid, the Set Link Up bit of the Device Control Register (CTRL.SLU) must be set. Link up provides a useful indication of whether something is attached to the port. Successful negotiation of features/link parameters results in link activity. The link startup process (and consequently the duration for this activity after reset) can be several 100's of ms. When the internal PHY is used, this reflects whether the PHY's LINK indication is present. When the SerDes or SGMII interface is used, this indicates loss-of-signal; if Auto-Negotiation is also enabled, this can also indicate successful auto-negotiation. Refer to Section 3.5.4 for more details.
LAN ID	3:2	0b	LAN ID Provides software a mechanism to determine the LAN identifier for the MAC. 00b = LAN 0. 01b = LAN 1.
TXOFF	4	X	Transmission Paused This bit indicates the state of the transmit function when symmetrical flow control has been enabled and negotiated with the link partner. This bit is set to 1b when transmission is paused due to the reception of an XOFF frame. It is cleared (0b) upon expiration of the pause timer or the receipt of an XON frame.
Reserved	5	X	Reserved
SPEED	7:6	X	Link Speed Setting Reflects the speed setting of the MAC and/or link when it is operating in 10/100/1000BASE-T mode (internal PHY). When the MAC is operating in 10/100/1000BASE-T mode with the internal PHY, these bits normally reflect the speed of the actual link, negotiated by the PHY and link partner and reflected internally from the PHY to the MAC (SPD_IND). These bits also might represent the speed configuration of the MAC only, if the MAC speed setting has been forced via software (CTRL.SPEED) or if MAC auto-speed detection is used. If Auto-Speed Detection is enabled, the 82576's speed is configured only once after the LINK signal is asserted by the PHY. 00b = 10 Mb/s. 01b = 100 Mb/s. 10b = 1000 Mb/s. 11b = 1000 Mb/s.
ASDV	9:8	X	Auto-Speed Detection Value Speed result sensed by the 82576's MAC auto-detection function. These bits are provided for diagnostics purposes only. The ASD calculation can be initiated by software writing a logic 1b to the CTRL_EXT.ASDCHK bit. The resultant speed detection is reflected in these bits.
PHYRA	10	1b	PHY Reset Asserted This read/write bit is set by hardware following the assertion of a PHY reset; it is cleared by writing a 0b to it. This bit is also used by firmware indicating a required initialization of the 82576's PHY.

Field	Bit(s)	Initial Value	Description
Reserved	13:11	0x0	Reserved
Num VFs	17:14	0x0	Reflects the value of the Num VFs in the IOV capability structure.
IOV mode	18	0b	Reflects the value of the VF enable (VFE) bit in the IOV capability structure.
GIO Master Enable Status	19	1b	Cleared by the 82576 when the GIO Master Disable bit is set and no master requests are pending by this function. Set otherwise. Indicates that no master requests are issued by this function as long as the GIO Master Disable bit is set.
Reserved	30:20	0x0	Reserved
DMA clock gating Enable	31	1b[1]	DMA clock gating Enable bit loaded from the EEPROM- indicates the device support gating of the DMA clock.

[1] If the signature bits of the EEPROM's Initialization Control Word 1 match (01b), this bit is read from the EEPROM.

8.2.3 Extended Device Control Register - CTRL_EXT (0x00018; R/W)

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.
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.
Reserved	5:4	00b	Reserved. Reads as 00b.
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.
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.
Reserved	9:8	0b[1]	Reserved Formally used as SDP5 and SDP4 pin input/output direction control, respectively.
SDP2_IODIR	10	0b[1]	SDP2 Pin Direction. 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.
SDP3_IODIR	11	0b[1]	SDP3 Pin Direction. 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.
ASDCHK	12	0b	ASD Check Initiates an Auto-Speed-Detection (ASD) sequence to sense the frequency of the PHY receive clock (RX_CLK). The results are reflected in STATUS.ASDV. This bit is self-clearing.

Field	Bit(s)	Initial Value	Description
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 the 82576 functions should be disabled prior to setting this bit. This bit is self-clearing.
PFRSTD (SC)	14	0b	PF reset Done When set, the RSTI bit in all the VFMailbox regs is cleared and the RSTD bit in all the VFMailbox regs is set.
SPD_BYPS	15	0b	Speed Select Bypass When set to 1b, all speed detection mechanisms are bypassed, and the 82576 is immediately set to the speed indicated by CTRL.SPEED. This provides a method for software to have full control of the speed settings of the 82576 and when the change takes place, by overriding the hardware clock switching circuitry.
NS_DIS	16	0	No Snoop Disable When set to 1b, the 82576 does not set the no snoop attribute in any PCIe packet, independent of PCIe configuration and the setting of individual no snoop enable bits. When set to 0b, behavior of no snoop is determined by PCIe configuration and the setting of individual no snoop enable bits.
RO_DIS	17	0b	Relaxed Ordering Disabled When set to 1b, the 82576 does not request any relaxed ordering transactions regardless of the state of bit 4 in the PCIe device control register (offset oxA8). When this bit is cleared and bit 4 of the PCIe device control register is set, the 82576 requests relaxed ordering transactions as provided by registers RXCTL and TXCTL (per queue and per flow).
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 5.5.4.
L1 Enable	19	0b[1]	When set, enables L1 indication.
PHY Power Down Enable	20	1b[1]	When set, enables the PHY to enter a low-power state as described in Section 5.4.3.
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). 01b = Reserved. 10b = SGMII. 11b = Internal SerDes interface.
Reserved	24	0b	Reserved
I2C Enabled	25	0b[1]	Enable I2C This bit enables the I^{2}C bus that can be used to access SFP modules in the EEPROM. If cleared, the I^{2}C pads are isolated and accesses through I2CCMD are ignored.
Extended VLAN	26	0b[1]	Extended VLAN When set, all incoming Rx packets are expected to have at least one VLAN with the Ether type as defined in VET.EXT_VET that should be ignored. The packets can have a second VLAN that should be used for all filtering purposes. All Tx packets are expected to have at least one VLAN added to them by the host. In the case of an additional VLAN request (VLE) the second VLAN is added after the VLAN is added by the host. This bit is reset only by a power up reset or by an EEPROM full auto load and should only be changed while Tx and Rx processes are stopped.
RS_RT_EN	27	0b	0 = Rate Scheduler are not reset at link speed change. 1 = Rate Scheduler are reset at link speed change.

Field	Bit(s)	Initial Value	Description
DRV_LOAD	28	0b	Driver Loaded This bit should be set by the driver after it is 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.
Reserved	29	0b	Reserved
Reserved	31:30	0b	Reserved

1. These bits are read from the EEPROM.

This register provides extended control of the 82576's functionality beyond that provided by the Device Control register (CTRL).

The 82576 allows up to four externally controlled interrupts. All software-definable pins, these can be mapped for use as GPI interrupt bits. Mappings are enabled by the SDPx_GPIEN bits only when these signals are also configured as inputs via SDPx_IODIR. When configured to function as external interrupt pins, a GPI interrupt is generated when the corresponding pin is sampled in an active-high state.

The bit mappings are shown in the table bellow for clarity.

Table 8-8. Mappings for SDI Pins Used as GPI

SDP Pin Used as GPI	CTRL_EXT Field Settings		Resulting ICR Bit (GPI)
	Direction	Enable as GPI interrupt	
3	SDP3_IODIR	SDP3_GPIEN	14
2	SDP2_IODIR	SDP2_GPIEN	13
1	SDP1_IODIR	SDP1_GPIEN	12
0	SDP0_IODIR	SDP0_GPIEN	11

Note: If software uses the EE_RST function and desires to retain 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 space BAR settings. The changes to the BARs are not visible unless the system reboots and the BIOS are allowed to re-map them.

The SPD_BYPS bit performs a similar function to the CTRL.FRCSPD bit in that the 82576'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 82576's clock switching circuitry performs the change.

8.2.4 MDI Control Register - MDIC (0x00020; R/W)

Software uses this register to read or write Management Data Interface (MDI) registers in the internal PHY or an external SGMII PHY.

See Section 3.5.2.2.1 for details of the usage of this register

The PHY registers accessible through the MDIC register are described in Section 8.24.

Field	Bit(s)	Initial Value	Description
DATA	15:0	X	Data In a Write command, software places the data bits and the MAC shifts them out to the PHY. In a Read command, the MAC 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
OP	27:26	0b	Opcode 01b = MDI Write 10b = MDI Read All other values are reserved.
R (RWS)	28	1b	Ready Bit Set to 1b by the 82576 at the end of the MDI transaction (for example, indication of a Read or Write completion). It should be reset to 0b by software at the same time the command is written.
I	29	0b	Interrupt Enable When set to 1b by software, it causes an Interrupt to be asserted to indicate the end of an MDI cycle.
E (RWS)	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.
Destination	31	0b	Destination 0b = The transaction is to the internal PHY. 1b = The transaction is directed to the I^{2}C Interface.

8.2.5 SerDes ANA - SERDESCTL (0x00024; R/W)

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.

8.2.6 Copper/Fiber Switch Control - CONNSW (0x00034; R/W)

Field	Bit(s)	Initial Value	Description
AUTOSENSE_EN	0	0b	Auto Sense Enable When set, the auto sense mode is active. In this mode the non-active link is sensed by hardware as follows PHY Sensing: The electrical idle detector of the receiver of the PHY is activated while in SerDes or SGMII mode. SerDes sensing: The electrical idle detector of the receiver of the SerDes is activated while in internal PHY mode, assuming the ENRGSRC bit is cleared If energy is detected in the non active media, the OMED bit in the ICR register is set and this bit is cleared. This includes the case where energy was present at the non-active media when this bit is being set.
AUTOSENSE_ CONF	1	0b	Auto Sense Config Mode This bit should be set during the configuration of the PHY/SerDes towards the activation of the auto-sense mode. While this bit is set, the PHY/SerDes is active even though the active link is set to SerDes or SGMII/PHY. Energy detection while this bit is set is not reflected to the OMED interrupt.
ENRGSRC	2	0b[1]	SerDes Energy Detect Source If set, the OMED interrupt cause is set after asserting the external signal detect pin. If cleared, the OMED interrupt cause is set after exiting from electrical idle of the SerDes receiver. This bit also defines the source of the signal detect indication used to set link up while is SerDes mode.
Reserved	8:3	0x0	Reserved
SerDesD (RO)	9	X	SerDes Signal Detect Indication Indicates the SerDes signal detect value according to the selected source (either external or internal). Valid only if LINK_MODE is SerDes or SGMII.
PHYSD (RO)	10	X	PHY Signal Detect Indication Valid only if LINK_MODE is the PHY and the receiver is not in electrical idle.
Reserved	31:11	0x0	Reserved

[1] Words 0x24 and 0x14 (bit 15) in the EEPROM defines the default of the ENRGSRC bit in this register for LAN0 and LAN1 respectively.

8.2.7 VLAN Ether Type - VET (0x00038; 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 0x8100. For VLAN transmission the upper byte is first on the wire (VET[15:8]).

Field	Bit(s)	Initial Value	Description
VET	15:0	0x8100	VLAN EtherType Should be programmed with 0x8100.
VET EXT	31:16	0x8100	External VLAN Ether Type.

8.2.8 LED Control - LEDCTL (0x00E00; RW)

This register controls the setup of the LEDs. See Section 7.5.1 for details of the MODE fields encoding.

Field	Bit(s)	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.
LED_PCI_MODE	4	0b	0b = Use LEDs as defined in the other fields of this register. 1b = Use LEDs to indicate PCI-E Lanes Idle status in SDP mode (only when the led_mode is set to 0x8 – SDP mode) For Port 0 LED0 3-0 indicates RX lanes 3- 0 Electrical Idle status For Port 1 LED1 3-0 indicates TX lanes 3- 0 Electrical Idle status
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 FILTER ACTIVITY# indication.
Reserved	12	0b	Reserved Read-only as 0b. Write as 0b for future compatibility.
Reserved	13	0b	Reserved
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	0b	Reserved Read-only as 0b. Write as 0b for future compatibility.
Reserved	21	0b	Reserved
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.
Reserved	28	0b	Reserved Read-only as 0b. Write as 0b for future compatibility.
Reserved	29	0b	Reserved
LED3_IVRT	30	0b[1]	LED3/LINK1000# Invert
LED3_BLINK	31	0b[1]	LED3/LINK1000# Blink

[1] These bits are read from EEPROM words 0x1C and 0x1F for port A and from EEPROM words 0x2B and 0x2C for port B.

8.3 Packet Buffers Control Register Descriptions

These registers set the on-chip receive, transmit & loopback storage allocation. The partitioning size is 1 KB.

Note: Programming these registers automatically initialize internal packet-buffer RAM pointers.

For best performance, the transmit buffer allocation should be set to accept two full-sized packets (For good 9 KB jumbo frame performance, the transmit allocation should be a minimum of 18 KB).

Transmit packet buffer size should be configured to be more than 8 KB.

8.3.1 RX PB Size - RXPBS (0x2404; RW)

Field	Bit(s)	Initial Value	Description
RXpbsize0	6:0	0x40	RX Packet buffer size. Value is in Kbytes. The default is 64K.
Reserved	31:7	0x0	Reserved.

8.3.2 TX PB Size - TXPBS (0x3404; RW)

Field	Bit(s)	Initial Value	Description
TXpbsize0	5:0	0x28	TX Packet buffer size. Value is in Kbytes. The default is 40K.
Reserved	31:6	0x0	Reserved.

8.3.3 Switch PB Size - SWPBS (0x3004; RW)

Field	Bit(s)	Initial Value	Description
SWpbsize0	4:0	0x14	Switch Packet buffer size. Value is in Kbytes. The default is 20K.
Reserved	31:5	0x0	Reserved.

8.3.4 Tx Packet Buffer Wrap Around Counter - PBTWAC (0x34e8; RO)

Field	Bit(s)	Initial Value	Description
WAC0	2:0	0x0	Reflects the wrap around of the entire Packet buffer.
TC0E	3	1b	Tx Packet buffer is empty
Reserved	31:4	0x0	Reserved.

8.3.5 Rx Packet Buffer Wrap Around Counter - PBRWAC (0x24e8; RO)

Field	Bit(s)	Initial Value	Description
WAC0	2:0	0x0	Reflects the wrap around of the entire Packet buffer.
TC0E	3	1b	Rx Packet buffer is empty
Reserved	31:4	0x0	Reserved.

8.3.6 Switch Packet Buffer Wrap Around Counter - PBSWAC (0x30e8; RO)

Field	Bit(s)	Initial Value	Description
WAC0	2:0	0x0	Reflects the wrap around of the entire Packet buffer.
TC0E	3	1b	switch Packet buffer is empty
Reserved	31:4	0x0	Reserved.

8.4 EEPROM/Flash Register Descriptions

Note: As the EEPROM and the flash are shared resources between the two ports and the manageability firmware, access to these resources should be coordinated using the semaphore mechanism. See Section 4.6 for details.

8.4.1 EEPROM/Flash Control Register - EEC (0x00010; R/W)

This register provides software direct access to the EEPROM. Software can control the EEPROM by successive writes to this register. Data and address information is clocked into the EEPROM by software toggling the EE_SK and EE_DI bits (0 and 2) of this register with EE_CS set to 0b. Data output from the EEPROM is latched into the EE_DO bit (bit 3) via the internal 62.5 MHz clock and can be accessed by software via reads of this register.

Note: Attempts to write to the Flash device via PCIe BAR or via I/O access when writes are disabled (FWE is not equal to 10b) should not be attempted. Behavior after such an operation is undefined and can result in component and/or system hangs. Bit banging access to the flash via FLA register is not protected by this field.

Field	Bit(s)	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.

Field	Bit(s)	Initial Value	Description
EE_DO (RO)	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.
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 (RO)	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 (RO)	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.
EE_ADDR_SIZE	10	0b	EEPROM Address Size This field defines the address size of the EEPROM. This bit is set by the EEPROM size auto-detect mechanism. If no EEPROM is present or the signature is not valid, a 16-bit address is assumed. 0b = 8- and 9-bit. 1b = 16-bit.
EE_SIZE (RO)	14:11[1]	0010b	EEPROM Size This field defines the size of the EEPROM: Field ValueEEPROM SizeEEPROM Address Size 0000b 128 bytes - 1 byte 0001b 256 bytes - 1 byte 0010b 512 bytes - 1 byte 0011b 1 Kbytes - 2 bytes 0100b 2 Kbytes - 2 bytes 0101b 4 Kbytes - 2 bytes 0110b 8 Kbytes - 2 bytes 0111b 16 Kbytes - 2 bytes 1000b 32 Kbytes - 2 bytes Reserved1011b - 1111b Reserved
Reserved	31:15	0b	Reserved Reads as 0b.

[1] These bits are red from the EEPROM.

8.4.2 EEPROM Read Register - EERD (0x00014; RW)

This register is used by software to cause the 82576 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 82576 reads the word from the EEPROM and places it in the *Read Data* field, setting the *Read Done* field to 1b. Software can poll this register, looking for a 1b in the *Read Done* field, and then using the value in the *Read Data* field.

When this register is used to read a word from the EEPROM, that word does not influence any of the 82576's internal registers even if it is normally part of the auto-read sequence.

Field	Bit(s)	Initial Value	Description
START	0	0b	Start Read Writing a 1b to this bit causes the EEPROM to read a (16-bit) word at the address stored in the EE_ADDR field and then storing the result in the EE_DATA field. This bit is self-clearing.
DONE (RO)	1	0b	Read Done Set to 1b when the EEPROM read completes. Set to 0b when the EEPROM read is not completed. Writes by software are ignored. Reset by setting the START bit.
ADDR	15:2	0x0	Read Address This field is written by software along with *Start Read* to indicate the word to read.
DATA (RO)	31:16	X	Read Data. Data returned from the EEPROM read.

8.4.3 Flash Access - FLA (0x0001C; R/W)

This register provides software direct access to the Flash. Software can control the Flash by successive writes to this register. Data and address information is clocked into the Flash by software toggling the FL_SCK bit (bit 0) of this register with FL_CE set to 1b. Data output from the Flash is latched into the FL_SO bit (bit 3) of this register via the internal 125 MHz clock and can be accessed by software via reads of this register.

Note: In the 82576, the Flash Access register is only reset at Internal_Power_On_Reset and not as legacy devices at a software reset.

Field	Bit(s)	Initial Value	Description
FL_SCK	0	0b	Clock Input to the Flash When FL_GNT is 1b, the FL_SCK out signal is mapped to this bit and provides the serial clock input to the Flash device. Software clocks the Flash memory via toggling this bit with successive writes.
FL_CE	1	0b	Chip 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.
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_SO 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. If this bit is set by one of the functions, it is also reflected in the other one.
Reserved	29:7	0b	Reserved Reads as 0b.
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 is sent to the Flash component only if R.FWE field is cleared. This bit is automatically cleared and read as 0b.

8.4.4 Flash Opcode - FLASHOP (0x0103C; R/W)

This register enables the host or the firmware to define the op-code used in order to erase a sector of the flash or the complete flash. This register is reset only at Internal_Power_On_Reset assertion.

This register is common to both ports and to the manageability and should be programmed according to the parameters of the flash used.

Note: The default values fit to Atmel* Serial Flash Memory devices.

Field	Bit(s)	Initial Value	Description
DERASE	7:0	0x0062	Flash Device Erase Instruction The op-code for the Flash erase instruction.
SERASE	15:8	0x0052	Flash Block Erase Instruction The op-code for the Flash block erase instruction. Relevant only to Flash access by manageability.
Reserved	31:16	0x0	Reserved

8.4.5 EEPROM Diagnostic - EEDIAG (0x01038; RO)

This register reflects the values of EEPROM bits influencing the hardware that are not reflected otherwise.

Field	Bit(s)	Initial Value	Description
LAN0 Disable Strap Behavior	0	0b	Reflects the inverse of bit 13 in EEPROM word 0x20 controlling behavior of disabling strap for LAN0.
LAN1 Disable Strap Behavior	1	0b	Reflects the inverse of bit 13 in EEPROM word 0x10 controlling behavior of disabling strap for LAN1.
LAN1 Disable	2	0b	Reflects bit 11 in EEPROM word 0x10 controlling the disabling of LAN1 as PCIe.
LAN1 PCI Disable	3	0b	Reflects bit 10 in EEPROM word 0x10 controlling the disabling of LAN1 as PCIe.

EEPROM Deadlock Release Enable	4	0b	Reflects bit 5 in EEPROM word 0x0A controlling the EEPROM deadlock release enable.
Dynamic IDDQ Enable	5	0b	Reflects bit 15 in EEPROM word 0x1E controlling the dynamic IDDQ enable.
PLL Shutdown Enable	6	0b	Reflects bit 4 in EEPROM 0x0F controlling the PLL shutdown enable control.
PLL Switch	7	0b	Reflects bit 5 in EEPROM word 0x21 controlling the timing of the switch to PLL clock.
NC- SI Clock Out	8	0b	Reflects the clock out setting in bit 13 of EEPROM word 0x21.
NC-SI Clock and I/O Pads Strength	10:9	00b	Reflects the clock and I/O pad drive strength settings in bits 15:14 of EEPROM word 0x21.
SDP_IDDQ_EN	11	0b	Reflects SDP behavior in the dynamic IDDQ setting in bit 6 of EEPROM word 0xA.
EEPROM Parallel State	13:12	X	State of the EEPROM parallel access arbitration state machine.
EEPROM Serial State	15:14	X	State of the EEPROM serial access arbitration state machine.
Flash Serial State	17:16	X	State of the Flash serial access arbitration state machine.
Flash Read Data State	19:18	X	State of the Flash read data bus arbitration state machine.
Flash Parallel State	22:20	X	State of the Flash parallel access arbitration state machine.
Reserved	30:23	0x0	Reserved
Deadlock Release	31	X	Indicates a deadlock condition was detected in the EEPROM and the current grant was released.

8.4.6 EEPROM Auto Read Bus Control - EEARBC (0x01024; R/W)

In EEPROM-less implementations, this register is used to program the 82576 the same way it should be programmed if an EEPROM was present. See Section 3.3.1.6.1 for details of this register usage.

This register is common to both functions and should be accessed only after the coordination with the other port.

Field	Bit(s)	Initial Value	Description
VALID_CORE0	0	0b	Valid Write Active to Core 0 Write strobe to Core 0. Firmware/software sets this bit for write access. Software should clear this bit to terminate the write transaction.
VALID_CORE1	1	0b	Valid Write Active to Core 1 Write strobe to Core 1. Firmware/software sets this bit for write access. Software should clear this bit to terminate the write transaction.
VALID_ COMMON	2	0b	Valid Write Active to Common Write strobe to Common. Firmware/software sets this bit for write access. Software should clear this bit to terminate the write transaction.
Reserved	3	0b	Reserved Reads as 0b.

ADDR	12:4	0x0	Write Address This field specifies the 9-bit lower bit of the word address of the EEPROM data.
Reserved	15:13	000b	Reserved Reads as 0b.
DATA	31:16	0x0	Data written into the EEPROM auto read bus.

1. More than one valid bit can be set for write accesses. This results in writing the specific address to more than one destination.
2. Not all EEPROM addresses are part of the auto read. By using this register software can write to the hardware registers that are configured during auto read only.
3. Write access to address 0x12 in the EEPROM is protected if a valid EEPROM exist. This limitation protects the secured EEPROM mechanism.

8.4.7 VPD diagnostic register -VPDDIAG (0x1060; RO)

This register stores the VPD parameters as parsed by the auto-load process. This register is used for debug only.

Field	Bit(s)	Initial Value	Description
Valid	0	X	VPD structure valid
Reserved	4:1	X	Reserved
RD Tag	13:5	X	Offset of the Read tag in VPD relative to the start of VPD (in bytes).
WR Tag	22:14	X	Offset of the Write tag in VPD relative to the start of VPD (in bytes).
End Tag	31:23	X	Offset of the End tag in VPD relative to the start of VPD (in bytes).

8.4.8 MNG-EEPROM CSR I/F

The following registers are reserved for Firmware access to the EEPROM and are not writable by the host.

8.4.8.1 MNG EEPROM Control Register - EEMNGCTL (0x1010; RO)

Field	Bit(s)	Initial Value	Description
ADDR	14:0	0x0	Address - This field is written by MNG along with Start Read or Start write to indicate the EEPROM address to read or write.
START	15	0b	Start - Writing a 1b to this bit causes the EEPROM to start the read or write operation according to the write bit.
WRITE	16	0b	Write - This bit tells the EEPROM if the current operation is read or write: 0b = read 1b = write
EEBUSY	17	0b	EEPROM Busy - This bit indicates that the EEPROM is busy processing an EEPROM transaction and shouldn't be accessed.

CFG_DONE 0	18	0b	MNG Configuration cycle is done for port 0 – This bit indicates that the MNG configuration cycle (configuration of SerDes, PHY, PCIe and PLLs) is done for port 0. This bit is set to 1b by MNG firmware to indicate configuration done and cleared at by hardware on any of the reset sources that cause the firmware to init the PHY. Write 0b by the firmware does not affect the state of this bit. Note: Port 0 driver should not try to access the PHY for configuration before this bit is set. Note: When the LAN Function Select bit in the EEPROM (word 0x21 bit 12 - See Section 6.2.20), this bit indicates that the MNG Configuration cycle is done for port **1.**
CFG_DONE 1	19	0b	MNG Configuration cycle is done for port 1– This bit indicates that the MNG configuration cycle (configuration of SerDes, PHY, PCIe and PLLs) is done for port 1. This bit is set to 1b by MNG firmware to indicate configuration done and cleared at by hardware on any of the reset sources that cause the firmware to init the PHY. Write 0b by the firmware does not affect the state of this bit. Note: Port 1 driver should not try to access the PHY for configuration before this bit is set. Note: When the LAN Function Select bit in the EEPROM (word 0x21 bit 12 - See Section 6.2.20), this bit indicates that the MNG Configuration cycle is done for port **0.**
Reserved	30:20	0x0	Reserved
DONE	31	1b	Transaction Done - This bit is cleared after Start Write or Start Read bit is set by the MNG and is set back again when the EEPROM write or read transaction is done.

8.4.8.2 MNG EEPROM Read / Write data - EEMNGDATA (0x1014; RO)

Field	Bit(s)	Initial Value	Description
WRDATA	15:0	0x0	Write Data **-** Data to be written to the EEPROM.
RDDATA	31:16	–	Read Data **-** Data returned from the EEPROM read.

8.5 Flow Control Register Descriptions

8.5.1 Flow Control Address Low - FCAL (0x00028; RO)

Flow control packets are defined by 802.3X to be either a unique multicast address or the station address with the Ether Type field indicating PAUSE. The FCA registers provide the value hardware uses to compare incoming packets against, to determine that it should PAUSE its output.

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. The complete flow control multicast address is: 0x01_80_C2_00_00_01; where 0x01 is the first byte on the wire, 0x80 is the second, etc.

Note: Any packet matching the contents of {FCAH, FCAL, FCT} when CTRL.RFCE is set is acted on by the 82576. Whether flow control packets are passed to the host (software) depends on the state of the RCTL.DPF bit and whether the packet matches any of the normal filters

Field	Bit(s)	Initial Value	Description
FCAL	31:0	0x00C28001	Flow Control Address Low

8.5.2 Flow Control Address High - FCAH (0x0002C; RO)

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}.

The complete flow control multicast address is: 0x01_80_C2_00_00_01; where 0x01 is the first byte on the wire, 0x80 is the second, etc.

Field	Bit(s)	Initial Value	Description
FCAH	15:0	0x0100	Flow Control Address High Should be programmed with 0x01_00.
Reserved	31:16	0b	Reserved Reads as 0b.

8.5.3 Flow Control Type - FCT (0x00030; R/W)

This register contains the type field that hardware matches to recognize a flow control packet. Only the lower 16 bits of this register have meaning. This register should be programmed with 0x88_08. The upper byte is first on the wire FCT[15:8].

Field	Bit(s)	Initial Value	Description
FCT	15:0	0x8808	Flow Control Type
Reserved	31:16	0b	Reserved Reads as 0b.

8.5.4 Flow Control Transmit Timer Value - FCTTV (0x00170; R/W)

The 16-bit value in the TTV field is inserted into a transmitted frame (either XOFF frames or any PAUSE frame value in any software transmitted packets). It counts in units of slot time of 64 bytes. If software needs to send an XON frame, it must set TTV to 0b prior to initiating the PAUSE frame.

Field	Bit(s)	Initial Value	Description
TTVTC0	15:0	X	Transmit Timer Value These bits are included in the XOFF frame
Reserved	31:16	0b	Reserved

8.5.5 Flow Control Receive Threshold Low - FCRTL0 (0x02160; R/W)

This register contains the receive threshold used to determine when to send an XON packet The complete register reflects the threshold in units of bytes. The lower 4 bits must be programmed to 0b (16 byte granularity). Software must set XONE to enable the transmission of XON frames. Each time hardware crosses the receive-high threshold (becoming more full), and then crosses the receive-low threshold and XONE is enabled (1b), hardware transmits an XON frame. When XONE is set, the *RTL* field should be programmed to at least 1b (at least 16 bytes).

Flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the 82576 is manually configured, flow control operation is determined by the *CTRL.RFCE* and *CTRL.TFCE* bits.

Field	Bit(s)	Initial Value	Description
Reserved	3:0	0000b	Reserved Must be written with 0b.
RTL	15:4	0x0	Receive Threshold Low. FIFO low water mark for flow control transmission. An XON packet is sent if the occupied space in the packet buffer is smaller or equal than this watermark. This field is in 16 bytes granularity.
Reserved	30:16	0x0	Reserved Should be written with 0b for future compatibility. Reads as 0b.
XONE	31	0b	XON Enable 0b = Disabled. 1b = Enabled.

8.5.6 Flow Control Receive Threshold High - FCRTH0 (0x02168; R/W)

This register contains the receive threshold used to determine when to send an XOFF packet. The complete register reflects the threshold in units of bytes. This value must be at maximum 48 bytes less than the maximum number of bytes allocated to the Receive Packet Buffer (PBA, RXA), and the lower 4 bits must be programmed to 0b (16 byte granularity). The value of RTH should also be bigger than FCRTL.RTL. 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 82576 is manually configured, flow control operation is determined by the *CTRL.RFCE* and *CTRL.TFCE* bits.

Field	Bit(s)	Initial Value	Description
Reserved	3:0	000b	Reserved Must be written with 0b.
RTH	15:4	0x0	Receive Threshold High FIFO high water mark for flow control transmission. An XOFF packet is sent if the occupied space in the packet buffer is bigger or equal than this watermark. This field is in 16 bytes granularity.
Reserved	31:16	0x0	Reserved Must be set to 0b.

8.5.7 Flow Control Refresh Threshold Value - FCRTV (0x02460; R/W)

Field	Bit(s)	Initial Value	Description
FC_refresh_th	15:0	0x0	Flow Control Refresh Threshold This value indicates the threshold value of the flow control shadow counter; when the counter reaches this value, and the conditions for PAUSE state are still valid (buffer fullness above low threshold value), a PAUSE (XOFF) frame is sent to link partner. If this field contains zero value, the Flow Control Refresh is disabled. Note: This register controls both TCs.
Reserved	31:16	-	Reserved

8.5.8 Flow Control Status - FCSTS0 (0x2464; RO)

This register describes the status of the flow control machine.

Field	Bit(s)	Initial Value	Description
Flow_control state	0	0b	Flow control state machine signal 0b = XON 1b = XOFF
Above high	1		The size of data in the memory is above the high threshold
Below low	2		The size of data in the memory is below the low threshold
Reserved	15:3	0x0	Reserved
Refresh counter	31:16	0x0	Flow control refresh counter

8.6 PCIe Register Descriptions

8.6.1 PCIe Control - GCR (0x05B00; RW)

Field	Bit(s)	Initial Value	Description
Ignore RID	0	0x0	When set, the RID of all DMA accesses is the PF RID This bit should be kept at 0b for normal operation.
Reserved	1	0b	Reserved.
Firmware Self_Test_Enable	8	0b	When set, firmware should perform a self test. Reset at power good only.
Rx_L0s_Adjustment	9	1b	If set, the replay timer always adds the required L0s adjustment. When set to 0b, adds it only when Tx L0s are active. Reset at power good only.
Reserved	10	0	Reserved
Reserved	11	0	Reserved

Completion_Timeout_Value (RO or RW1)	15:12	0x0	Indicates the selected value for completion timeout. Decoding of this field depends on the PCIe capability version: Capability version = 1 (bits 13:12): 00b = 50 μs to 10 ms (default) 01b = 10 ms to 200 ms 10b = 200 ms to 4 s 11b = 4 s to 64 s Bits 15:14 are reserved Capability version = 2: 0000b = 50 μs to 50 ms 0001b = 50 μs to 100 μs 0010b = 1 ms to 10 ms 0011b = Reserved 0100b = Reserved 0101b = 16 ms to 55 ms 0110b = 65 ms to 210 ms 0111b = Reserved 1000b = Reserved 1001b = 260 ms to 900 ms 1010b = 1 s to 3.5 s 1011b = Reserved 1100b = Reserved 1101b = 4 s to 13 s 1110b = 17 s to 64 s 1111b = Reserved Reset at power good only.
Completion_Timeout_Resend	16	1b	When set, enables re-sending of a request once the completion timeout expired. 0b = Do not re-send request on completion timeout. 1b = Re-send request on completion timeout. This bit is used no matter which timeout mechanism is used. Reset at power good only.
Completion_Timeout_Disable (RO or RW1)	17	0b	Indicates if PCIe Completion Timeout is Supported 0b = Completion timeout enabled. 1b = Completion timeout disabled. Reset at power good only.
PCIe Capability Version (RO)	18	1b[1]	Reports the PCIe Capability Version Supported 0b = Capability version = 0x1. 1b = Capability version = 0x2.
Reserved	19	0	Reserved
PBA_CL_DEAS	20	0b	If cleared, PBA is cleared on de-assertion of MSI-X request.
hdr_log Inversion	21	0b	If set, the header log in error reporting is written as 31:0 to log1, 63:643 in log2. If not set, the header is written as 127:96 in log1 95:64 in log 2 ....
Reserved	23:22	1b	Reserved Must be set to 01b.
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).
L1_Entry_Latency (RO)	26:25	11b	Determines the idle time of the PCIe link in L0s state before initiating a transition to L1 state. Initial value is loaded from the EEPROM. 00b - 64 μs 01b - 256 μs 10b - 1 ms 11b - 4 ms
Reserved	31:27	0b	Reserved

1. The default value for this field is read from EEPROM word 0x18 bits 11:10. If these bits are set to 10b, then this fields is set to 1, otherwise it is set to zero.

8.6.2 IOV control- IOVCTL (0x05BBC; RW)

Field	Bit(s)	Initial Value	Description
Use VF Queues (WO)	0	0b	Should be set at least 1 ms after IOV was disabled, and before the PF reuses queues previously assigned to VFs. If the PF do not re-use the queues there is no need to set this bit.
Reserved	30:1	0x0	Reserved
Use VF Queues enable	31	1b	If set, then the Use VF Queues bit should be set before re-using the queues. If not set, VF enable in the config space should be cleared only after all VFs had be quiesced.

8.6.3 Function Tag - FUNCTAG (0x05B08; R/W)

Field	Bit(s)	Initial Value	Description
cnt_0_tag	4:0	0x0	Tag number for event 6/1D, if located in counter 0.
cnt_0_func	7:5	0x0	Function number for event 6/1D, if located in counter 0.
cnt_1_tag	12:8	0x0	Tag number for event 6/1D, if located in counter 1.
cnt_1_func	15:13	0x0	Function number for event 6/1D, if located in counter 1.
cnt_2_tag	20:16	0x0	Tag number for event 6/1D, if located in counter 2.
cnt_2_func	23:21	0x0	Function number for event 6/1D, if located in counter 2.
cnt_3_tag	28:24	0x0	Tag number for event 6/1D, if located in counter 3.
cnt_3_func	31:29	0x0	Function number for event 6/1D, if located in counter 3.

8.6.4 PCIe Statistics Control #1 - GSCL_1 (0x05B10; RW)

Field	Bit(s)	Initial Value	Description
GIO_COUNT_EN_0	0	b0	Enable PCIe Statistic Counter Number 0.
GIO_COUNT_EN_1	1	0b	Enable PCIe Statistic Counter Number 1.
GIO_COUNT_EN_2	2	0b	Enable PCIe Statistic Counter Number 2.
GIO_COUNT_EN_3	3	0b	Enable PCIe Statistic Counter Number 3.
Leaky_EN	7:4	0b	Enable Leaky Bucket for Counters 3:0
Reserved	26:8	0b	Reserved.
GIO_COUNT_TEST	27	0b	Test Bit Firmware counters for testability.
GIO_64_BIT_EN	28	0b	Enable two 64-bit counters instead of four 32-bit counters.

GIO_COUNT_ RESET	29	0b	Reset indication of PCIe statistic counters.
GIO_COUNT_ STOP	30	0b	Stop indication of PCIe statistic counters.
GIO_COUNT_ START	31	0b	Start indication of PCIe statistic counters.

8.6.5 PCIe Statistics Control #2 - GSCL_2 (0x05B14; RW)

This counter contains the mapping of an event (which counter counts what event).

Field	Bit(s)	Initial Value	Description
GIO_EVENT_ NUM_0	7:0	0b	Event number that counter 0 (GSCN_0) counts.
GIO_EVENT_ NUM_1	15:8	0b	Event number that counter 1 (GSCN_1) counts.
GIO_EVENT_ NUM_2	23:16	0b	Event number that counter 2 (GSCN_2) counts.
GIO_EVENT_ NUM_3	31:24	0b	Event number that counter 3 (GSCN_3) counts.

Table 8-9 lists the encoding of the events.

Table 8-9. Events Encoding

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 received. Counted: good completions of func#0

Table 8-9. Events Encoding (Continued)

Receive TLP Packets of function #1	1C	Each cycle, the counter increases by 1, if the packet was received. 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 PCIe cycles.
Bad TLP from LL	21	Each cycle, the counter increases by 1, if bad TLP is received (bad CRC, misplaced special char, bad link layer formatting of received TLPs).
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 Stops 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 stops 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 decreases to 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	
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)

Table 8-9. Events Encoding (Continued)

Received packets to func_logic.	36	Each 125 MHz cycle the counter increases by 1 Counted: memory, I/O, config (only good)
Average latency of read request – from initialization until end of completions. Estimated latency is ~5µSec.	40 + 41	The software selects the client that needs to be tested. The statistic counter counts the number of read requests of the required client. In addition, the accumulated time of all requests are saved in a time accumulator. The average time for read requests is [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 1µsec.	42 + 43	The software selects the client that needs to be tested. The statistic counter counts the number of read requests of the required client. In addition, the accumulated time of all RTTs is saved in a time accumulator. The average time for read requests is [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 selects the client that needs to be tested. The software programs the required threshold (in GSCL_4 – units of 96 ns). One statistic counter counts the time from the beginning of the request until end of completions. The other counter counts the number of events. If the time is above threshold – add 1 to the event counter. (Event 46 is for the counter).
Completion Latency above Threshold – for RTT	47 + 48	The software selects the client that needs to be tested. The software programs the required threshold (in GSCL_4 – units of 96 ns). One statistic counter counts the time from the beginning of the request until first completion arrival. The other counter counts the number of events. If the time is above threshold – add 1 to the event counter. (Event 48 is for the counter).
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

8.6.6 PCIe Statistics Control #3 - GSCL_3 (0x05B18; RW)

This counter holds the threshold values needed for some of the event counting. The event count increases only after the value passes the threshold boundary.

Field	Bit(s)	Initial Value	Description
GIO_FC_TH_0	11:0	0b	Threshold of flow control credits. Optional values: 0 - (256-1).
Reserved	15:12	0b	Reserved.
GIO_FC_TH_1	27:16	0b	Threshold of flow control credits. Optional values: 0 - (256-1).
Reserved	31:28	0b	Reserved.

8.6.7 PCIe Statistics Control #4 - GSCL_4 (0x05B1C; RW)

This counter holds the threshold values needed for some of the event counting. The event count increases only after the value passes the threshold boundary.

Field	Bit(s)	Initial Value	Description
GIO_COML_TH	9:0	0b	Completions latency threshold.
GIO_RB_TH	15:10	0b	Retry buffer threshold.
Reserved	31:16	0b	Reserved.

8.6.8 PCIe Statistic Control Register Leaky bucket timer- GSCL_LBT (0x5B90 + 4*n[n=0...3]; RW)

These registers control the operation of Leaky Bucket Counters 0 to 3 respectively.

Field	Bit(s)	Initial Value	Description
LBC threshold n	15:0	0x00	Threshold for the Leaky Bucket Counter n.
LBC timer n	31:16	0x00	Time period between decrementing the value in Leaky Bucket Counter n. The period is defined in µsec.

8.6.9 PCIe Counter #0 - GSCN_0 (0x05B20; RW)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0b	Event Counter

8.6.10 PCIe Counter #1 - GSCN_1 (0x05B24; RW)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0b	Event Counter

8.6.11 PCIe Counter #2 - GSCN_2 (0x05B28; RW)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0b	Event Counter

8.6.12 PCIe Counter #3 - GSCN_3 (0x05B2C; RW)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0b	Event Counter

8.6.13 Function Active and Power State to MNG - FACTPS (0x05B30; RO)

Firmware uses this register for configuration

Field	Bit(s)	Initial Value	Description
Func0 Power State	1:0	00b	Power state indication of Function 0 00b → DR 01b → D0u 10b →D0a 11b → D3
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. The LAN 0 enable is set by the LAN 0 Enable / TEST_POINT[2] strapping pin.
Func0 Aux_En	3	0b	Function 0 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
Reserved	5:4	0b	Reserved.
Func1 Power State	7:6	00b	Power state indication of Function 1 00b → DR 01b → D0u 10b → D0a 11b → D3
LAN1 Valid	8	0b	LAN 1 Enable When set to 0b, it indicates that the LAN 1 function is disabled. When the function is enabled, the bit is set to 1b. The LAN 1 enable is set by the LAN 1 Enable / TEST_POINT[3] strapping pin.
Func1 Aux_En	9	0b	Function 1 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
Reserved	28:10	0x0	Reserved

MNGCG	29	0b	MNG Clock Gated When set, indicates that the manageability clock is gated.
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 0x21.
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.

8.6.14 SerDes/CCM/PCIe CSR - GIOANACTL0 (0x05B34; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.15 SerDes/CCM/PCIe CSR - GIOANACTL1 (0x05B38; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.16 SerDes/CCM/PCIe CSR - GIOANACTL2 (0x05B3C; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.17 SerDes/CCM/PCIe CSR - GIOANACTL3 (0x05B40; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.18 SerDes/CCM/PCIe CSR - GIOANACTLALL (0x05B44; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.19 SerDes/CCM/PCIe CSR - CCMCTL (0x05B48; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.20 SerDes/CCM/PCIe CSR - SCCTL (0x05B4C; R/W)

Firmware uses this register for analog circuit configuration.

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 is completed this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

8.6.21 Mirrored Revision ID - MREVID (0x05B64; R/W)

Field	Bit(s)	Initial Value	Description
EEPROM RevID	7:0	0x0	Mirroring of Revision ID loaded from the EEPROM in PCI configuration space (from word 0x1E).
Default RevID	15:8	0x0	Mirroring of Default Rev ID before an EEPROM load. Set to 0b.
Reserved	31:16	0x0	Reserved

8.7 Semaphore registers

This section contains registers common to both cores used to coordinate between the two functions. The usage of these registers is described in Section 4.6

8.7.1 Software Semaphore - SWSM (0x05B50; R/W)

Field	Bit(s)	Initial Value	Description
SMBI (RS)	0	0x0	Software/Software 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 82576. This bit is cleared on GIO soft reset.
SWESMBI	1	0x0	Software/Firmware 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 than read it to see if it was set. If it was set, it means that the device driver can access the SW_FW_SYNC register. The device driver should clear this bit after modifying the SW_FW_SYNC register. Hardware clears this bit on GIO soft reset.
WMNG (SC)	2	0x0	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.
EEUR	3	0x0	EEPROM Update Request EEPROM request update from firmware. Software should clear this bit after the *FWSMFW_valid* bit is set.
Reserved	31:4	0x0	Reserved

8.7.2 Firmware Semaphore - FWSM (0x05B54; R/WS)

Field	Bit(s)	Initial Value	Description
EEP_FW_ Semaphore	0	0x0	Software/Firmware Semaphore Firmware should set this bit to 1b before accessing the SW_FW_SYNC register. If the software is using the SWSM does not lock the SW_FW_SYNC, firmware is able to set this bit to 1b. Firmware should set this bit to 0b after modifying the SW_FW_SYNC register.
FW_Mode	3:1	0x0	Firmware Mode Indicates the firmware mode as follows: 000b = No MNG. 001b = Reserved. 010b = PT mode. 011b = Reserved. 100b = Host Interface enable only.
Reserved	5:4	00b	Reserved
EEP_Reload_ Ind	6	0x0	EEPROM reloaded indication Set to 1b after firmware reloads the EEPROM. Cleared by firmware once the “Clear Bit” host command is received from host software.
Reserved	14:7	0x0	Reserved
FW_Val_Bit	15	0x0	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).
Reset_Cnt	18:16	0x0	Reset Counter Firmware increments the count at every reset.
Ext_Err_Ind	24:19	0x0	External error indication Firmware writes here the reason that the firmware has reset / clock gated. For example, EEPROM, flash, patch corruption, etc. Possible values: 0x00: No Error 0x01: Invalid EEPROM checksum 0x02: Unlocked secured EEPROM 0x03: Clock Off host command 0x04: Invalid Flash checksum 0x05: C0 checksum failed 0x06: C1 checksum failed 0x07: C2 checksum failed[1] 0x08: C3 checksum failed 0x09: TLB table exceeded 0x0A: DMA load failed 0x0B: Bad hardware version in patch load 0x0C: Flash device not supported 0x0D: Unspecified Error x03F: Reserved - max error value.
PCIe_ Config_Err_ Ind	25	0x0	PCIe configuration error indication Set to 1b by firmware when it fails to configure PCIe interface. Cleared by firmware upon successful configuration of PCIe interface.
PHY_SERDES0_ Config_Err_Ind	26	0x0	PHY/SerDes0 configuration error indication Set to 1b by firmware when it fails to configure LAN0 PHY/ SerDes. Cleared by firmware upon successful configuration of LAN0 PHY/ SerDes.

PHY_SERDES1_ Config_Err_Ind	27	0x0	PHY/SerDes1 configuration error indication Set to 1b by firmware when it fails to configure LAN1 PHY/ SerDes. Cleared by firmware upon successful configuration of LAN1 PHY/ SerDes.
Reserved	31:28	0x0	Reserved.

NOTES:

1. This register should be written only by the manageability firmware. The device driver should only read this register.
2. Firmware ignores the EEPROM semaphore in operating system hung states.
3. Bits 15:0 are cleared on firmware reset.

8.7.3 Software–Firmware Synchronization - SW_FW_SYNC (0x05B5C; RWS)

This register is intended to synchronize between software and firmware. This register is common to both ports 0 and 1.

Field	Bit(s)	Initial Value	Description
SW_EEP_SM	0	0b	When set to 1b, EEPROM access is owned by software
SW_PHY_SM0	1	0b	When set to 1b, PHY 0 access is owned by software
SW_PHY_SM1	2	0b	When set to 1b, PHY 1 access is owned by software
SW_MAC_CSR_ SM	3	0b	When set to 1b, software owns access to shared CSRs
SW_FLASH_SM	4	0	When set to 1b, software owns access to the flash.
Reserved	15:5	0x0	Reserved for future use
FW_EEP_SM	16	0b	When set to 1b, EEPROM access is owned by firmware
FW_PHY_SM0	17	0b	When set to 1b, PHY 0 access is owned by firmware
FW_PHY_SM1	18	0b	When set to 1b, PHY 1 access is owned by firmware
FW_MAC_CSR_ SM	19	0b	When set to 1b, firmware owns access to shared CSRs
FW_FLASH_SM	20	0	When set to 1b, firmware owns access to the flash.
Reserved	31:21	0x0	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 the bits on entry to D3 state.
- The firmware-controlled bits 31:16 are reset on Internal_Power_On_Reset and firmware reset.

8.8 Interrupt Register Descriptions

8.8.1 Extended Interrupt Cause - EICR (0x01580; RC/W1C)

This register contains the frequent interrupt conditions for the 82576. Each time an interrupt causing event occurs, the corresponding interrupt bit is set in this register. An interrupt is generated each time one of the bits in this register is set and the corresponding interrupt is enabled via the Interrupt Mask Set/Read register. The interrupt might be delayed by the selected Interrupt Throttling register.

Note that the software device driver cannot determine from the RxTxQ bits as to what was the cause of the interrupt:

- Receive Descriptor Write Back, Receive Descriptor Minimum Threshold hit, low latency interrupt for Rx, Transmit Descriptor Write Back.

Writing a 1b to any bit in the register clears that bit. Writing a 0b to any bit has no effect on that bit.

Register bits are cleared on register read.

Auto clear can be enabled for any or all of the bits in this register.

Table 8-10. EICR Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	15:0	0x0	Receive/Transmit Queue Interrupts One bit per queue or a bundle of queues, activated on receive/ transmit queue events for the corresponding bit, such as: Receive Descriptor Write Back, Receive Descriptor Minimum Threshold hit Transmit Descriptor Write Back. The mapping of actual queue to the appropriate RxTxQ bit is according to the IVAR registers.
Reserved	29:16	0x0	Reserved
TCP Timer	30	0b	TCP Timer Expired Activated when the TCP timer reaches its terminal count.
Other Cause	31	0b	Interrupt Cause Active Activated when any bit in the ICR register is set.

Table 8-11. EICR Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Indicates an interrupt cause mapped to MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved

Note: In IOV mode, only bit zero of this vector is available for the PF function.

8.8.2 Extended Interrupt Cause Set - EICS (0x01520; WO)

Software uses this register to set an interrupt condition. Any bit written with a 1b sets the corresponding bit in the Extended Interrupt Cause Read register. An interrupt is then generated if one of the bits in this register is set and the corresponding interrupt is enabled via the Extended Interrupt Mask Set/Read register. Bits written with 0b are unchanged.

Note: In order to set bit 31 of the EICR (Other Causes), the ICS and IMS registers should be used in order to enable one of the legacy causes.

Table 8-12. EICS Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	15:0	0x0	Sets to corresponding EICR RxTXQ interrupt condition.
Reserved	29:16	0x0	Reserved
TCP Timer	30	0b	Sets the corresponding EICR TCP interrupt condition.
Reserved	31	0b	Reserved

Table 8-13. EICS Register Bit Description - MSI-X Mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Sets to corresponding EICR bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved

8.8.3 Extended Interrupt Mask Set/Read - EIMS (0x01524; RWS)

Reading of this register returns which bits have an interrupt mask set. An interrupt in EICR is enabled if its corresponding mask bit is set to 1b and disabled if its corresponding mask bit is set to 0b. A PCI interrupt is generated each time one of the bits in this register is set and the corresponding interrupt condition occurs (subject to throttling). The occurrence of an interrupt condition is reflected by having a bit set in the Extended Interrupt Cause Read register.

An interrupt might be enabled by writing a 1b to the corresponding mask bit location (as defined in the EICR register) in this register. Any bits written with a 0b are unchanged. As a result, if software needs to disable an interrupt condition that had been previously enabled, it must write to the Extended Interrupt Mask Clear register rather than writing a 0b to a bit in this register.

Table 8-14. EIMS Register Bit Description - Non MSI-X Mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	15:0	0x0	Set Mask bit for the corresponding EICR RxTXQ interrupt.
Reserved	29:16	0x0	Reserved
TCP Timer	30	0b	Set Mask bit for the corresponding EICR TCP timer interrupt condition.
Other Cause	31	1b	Set Mask bit for the corresponding EICR other cause interrupt condition.

Table 8-15. EIMS Register Bit Description - MSI-X Mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Set Mask bit for the corresponding EICR bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved

8.8.4 Extended Interrupt Mask Clear - EIMC (0x01528; WO)

This register provides software a way to disable certain or all interrupts. Software disables a given interrupt by writing a 1b to the corresponding bit in this register.

On interrupt handling, the software device driver should set all the bits in this register related to the current interrupt request even though the interrupt was triggered by part of the causes that were allocated to this vector.

Interrupts are presented to the bus interface only when the mask bit is set to 1b and the cause bit is set to 1b. The status of the mask bit is reflected in the Extended Interrupt Mask Set/Read register and the status of the cause bit is reflected in the Interrupt Cause Read register.

Software blocks interrupts by clearing the corresponding mask bit. This is accomplished by writing a 1b to the corresponding bit location (as defined in the EICR register) of that interrupt in this register. Bits written with 0b are unchanged (their mask status does not change).

Table 8-16. EIMC Register Bit Description - Non MSI-X Mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	15:0	0x0	Clear Mask bit for the corresponding EICR RxTXQ interrupt.
Reserved	29:16	0x0	Reserved
TCP Timer	30	0b	Clear Mask bit for the corresponding EICR TCP timer interrupt.
Other Cause	31	1b	Clear Mask bit for the corresponding EICR other cause interrupt.

Table 8-17. EIMC Register Bit Description - MSI-X Mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	clear Mask bit for the corresponding EICR bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved

8.8.5 Extended Interrupt Auto Clear - EIAC (0x0152C; R/W)

This register is mapped like the EICS, EIMS, and EIMC registers, with each bit mapped to the corresponding MSI-X vector.

This register is relevant to MSI-X mode only, where read-to-clear can not be used, as it might erase causes tied to other vectors. If any bits are set in EIAC, the EICR register should not be read. Bits without auto clear set, need to be cleared with write-to-clear.

Note: EICR bits that have auto clear set are cleared by the internal emission of the corresponding MSI-X message even if this vector is disabled by the operating system.

The MSI-X message can be delayed by EITR moderation from the time the EICR bit is activated.

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Auto clear bit for the corresponding EICR bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved

8.8.6 Extended Interrupt Auto Mask Enable - EIAM (0x01530; R/W)

Each bit in this register enables clearing of the corresponding bit in EIMS following read- or write-to-clear to EICR or setting of the corresponding bit in EIMS following a write-to-set to EICS.

In MSI-X mode, this register controls which of the bits in EIMC to clear upon interrupt generation.

Table 8-18. EIAM Register Bit Description - Non MSI-X Mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	15:0	0x0	Auto Mask bit for the corresponding EICR RxTxQ interrupt.
Reserved	29:16	0x0	Reserved
TCP Timer	30	0b	Auto mask bit for the corresponding EICR TCP timer interrupt condition.
Other Cause	31	0b	Auto mask bit for the corresponding EICR other cause interrupt condition.

Table 8-19. EIAM Register Bit Description - MSI-X Mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Auto Mask bit for the corresponding EICR bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved

8.8.7 Interrupt Cause Read Register - ICR (0x01500; RC/W1C)

This register contains the interrupt conditions for the 82576 that are not present directly in the EICR. Each time an ICR interrupt causing event occurs, the corresponding interrupt bit is set in this register. The *EICR.Other* bit reflects the setting of interrupt causes from ICR as masked by the Interrupt Mask Set/Read register. Each time all un-masked causes in ICR are cleared, the *EICR.Other* bit is also cleared.

ICR bits are cleared on register read. Clear-on-read can be enabled/disabled through a general configuration register bit.

Auto clear is not available for the bits in this register.

In order to prevent unwanted LSC interrupts during initialization, software should disable this interrupt until the end of initialization.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Transmit Descriptor Written Back Set when the 82576 writes back a Tx descriptor to memory.
Reserved	1	0b	Reserved Should be set to 0b for compatibility.
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 LINK indication from the PHY (internal PHY mode).
RXSEQ	3	0b	Receive Sequence Error 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.
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
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.
RXDW	7	0b	Receiver Descriptor Write Back Set when the 82576 writes back an Rx descriptor to memory.
VMMB	8	0b	Set in IOV mode when a VF sends a message or an acknowledge of a message to the PF. Also set, when an FLR is asserted for one of the VFs.
Reserved	9	0b	Reserved
Reserved	10	0b	Reserved
GPI_SDP0	11	0b	General Purpose Interrupt on SDP0 If GPI interrupt detection is enabled on this pin (via CTRL_EXT), this interrupt cause is set when the SDP0 is sampled high.
GPI_SDP1	12	0b	General Purpose Interrupt on SDP1 If GPI interrupt detection is enabled on this pin (via CTRL_EXT), this interrupt cause is set when the SDP1 is sampled high.
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.
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.
PTRAP	15	0b	Probe trap interrupt - when set, the probe mode trap test mode trapped the requested event.
Reserved	17:16	000b	Reserved
MNG	18	0b	Manageability Event Detected Indicates that a manageability event happened. When the 82576 is at power down mode, the IPMI can generate a PME for the same events that would cause an interrupt when the 82576 is at the D0 state.
Reserved	19	0b	Reserved
OMED	20	0b	Other Media Energy Detect When in SerDes/SGMII mode, indicates that link status has changed on the 1000BASE-T PHY or when in 1000BASE-T PHY mode, there is a change in SerDes/SGMII link status.

Field	Bit(s)	Initial Value	Description
Reserved	21	0b	Reserved
FER	22	0b	Fatal Error This bit is set when a fatal error is detected in one of the memories
NFER	23	0b	Non Fatal Error This bit is set when a non fatal error is detected in one of the memories
CSRTO	24	0b	This bit is set when upon a CSR access time out indication.
SCE	25	0b	Storm Control Event This bit is set when multicast or broadcast storm control mechanism is activated or de-activated.
Software WD	26	0b	Software Watchdog This bit is set after a software watchdog timer times out.
Reserved	27	0b	Reserved
DOUTSYNC	28	0b	DMA Tx Detected out of Sync Situation Occurs when one of the queues used malformed Tx descriptors. In IOV systems, might indicate a malicious or buggy driver. **Note:** This bit should never rise during normal operation.
Reserved	29	0b	
TCP timer	30	00b	TCP timer interrupt
INTA	31	0	Interrupt Asserted: Indicates that the INT line is asserted. Can be used by driver in shared interrupt scenario to decide if the received interrupt was emitted by the 82576. This bit is not valid in MSI/MSI-X environments

8.8.8 Interrupt Cause Set Register - ICS (0x01504; WO)

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 8.8.7). 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 8.8.9).

Bits written with 0 are unchanged.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Sets the Transmit Descriptor Written Back Interrupt.
Reserved	1	-	Reserved
LSC	2	0b	Sets the Link Status Change Interrupt.
RXSEQ	3	0b	Sets the Receive Sequence Error Interrupt.
RXDMT0	4	0b	Sets the Receive Descriptor Minimum Threshold Hit Interrupt.
Reserved	5	0b	Reserved
RXO	6	0b	Sets the Receiver Overrun Interrupt.
RXDW	7	0b	Sets the Receiver Descriptor Write Back Interrupt.
VMMB	8	0b	Sets the VM mailbox interrupt.
Reserved	9	0b	Reserved
Reserved	10	0b	Reserved
GPI_SDP0	11	0b	Sets the General Purpose Interrupt, related to SDP0 pin.

Field	Bit(s)	Initial Value	Description
GPI_SDP1	12	0b	Sets the General Purpose Interrupt, related to SDP1 pin.
GPI_SDP2	13	0b	Sets the General Purpose Interrupt, related to SDP2 pin.
GPI_SDP3	14	0b	Sets the General Purpose Interrupt, related to SDP3 pin.
PTRAP	15	0	Set the Probe trap interrupt
Reserved	17:16	0b	Reserved.
MNG	18	0b	Sets the Management Event Interrupt.
Reserved	19	0b	Reserved.
OMED	20	0b	Sets the Other Media Energy Detected Interrupt.
Reserved	21	0b	Reserved
FER	22	0b	Sets the Fatal Error Interrupt.
NFER	23	0b	Sets the Non Fatal Error Interrupt.
CSRTO	24	0b	Sets the CSR access time out indication interrupt.
SCE	25	0b	Set the Storm Control Event Interrupt
Software WD	26	0b	Sets the Software Watchdog Interrupt.
Reserved	27	0b	Reserved.
DOUTSYNC	28	0b	Sets the DMA Tx Out of Sync Interrupt.
Reserved	29	0b	
TCP timer	30	0b	Sets the TCP timer interrupt.
Reserved	31	0b	Reserved.

8.8.9 Interrupt Mask Set/Read Register - IMS (0x01508; R/W)

Reading this register returns bits that have an interrupt mask set. 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 8.8.7).

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 8.8.10) rather than writing a 0b to a bit in this register.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Sets/Reads the mask for Transmit Descriptor Written Back Interrupt.
Reserved	1	-	Reserved
LSC	2	0b	Sets/Reads the mask for Link Status Change Interrupt.
RXSEQ	3	0b	Sets/Reads the mask for Receive Sequence Error Interrupt.
RXDMT0	4	0b	Sets/Reads the mask for Receive Descriptor Minimum Threshold Hit interrupt.
Reserved	5	0b	Reserved
RXO	6	0b	Sets/Reads the mask for Receiver Overrun interrupt.
RXDW	7	0b	Sets/Reads the mask for Receiver Descriptor Write Back interrupt.
VMMB	8	0b	Sets/Reads the mask for Mailbox interrupt.

Field	Bit(s)	Initial Value	Description
Reserved	9	0b	Reserved
Reserved	10	0b	Reserved
GPI_SDP0	11	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP0 pin.
GPI_SDP1	12	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP1 pin.
GPI_SDP2	13	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP2 pin.
GPI_SDP3	14	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP3 pin.
PTRAP	15	0	Set/read the mask for the Probe trap interrupt
Reserved	17:16	0b	Reserved.
MNG	18	0b	Sets/Reads the mask for Management Event Interrupt.
Reserved	19	0b	Reserved.
OMED	20	0b	Sets/Reads the mask for Other Media Energy Detected Interrupt.
Reserved	21	0b	Reserved
FER	22	0b	Sets/Reads the mask for the Fatal Error Interrupt.
NFER	23	0b	Sets/Reads the mask for the Non Fatal Error Interrupt.
CSRTO	24	0b	Sets/Reads the mask for the CSR access time out indication interrupt.
SCE	25	0b	Sets/Reads the mask for the Storm Control Event Interrupt.
Software WD	26	0b	Sets/Reads the mask for the Software Watchdog Interrupt.
Reserved	27	0b	Reserved.
OUTSYNC	28	0b	Sets/Reads the mask for DMA Tx Out of Sync Interrupt.
Reserved	29	0b	
TCP timer	30	0b	Sets/Reads the mask for TCP timer interrupt.
Reserved	31	0b	Reserved.

8.8.10 Interrupt Mask Clear Register - IMC (0x0150C; WO)

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 8.8.9), and the status of the cause bit is reflected in the Interrupt Cause Read Register (see Section 8.8.7). Reading this register returns the value of the IMS register.

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).

In interrupt handling, the software device driver should set all the bits in this register related to the current interrupt request, even though the interrupt was triggered by part of the causes that were allocated to this vector.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Clears the mask for Transmit Descriptor Written Back Interrupt.
Reserved	1	-	Reserved
LSC	2	0b	Clears the mask for Link Status Change Interrupt.
RXSEQ	3	0b	Clears the mask for Receive Sequence Error Interrupt.
RXDMT0	4	0b	Clears the mask for Receive Descriptor Minimum Threshold Hit Interrupt.
Reserved	5	0b	Reserved
RXO	6	0b	Clears the mask for Receiver Overrun Interrupt. Sets on Receive Data FIFO Overrun.
RXDW	7	0b	Clears the mask for Receiver Descriptor Write Back interrupt.
VMMB	8	0b	Clears the mask for VM mailbox interrupt.
Reserved	9	0b	Reserved
Reserved	10	0b	Reserved
GPI_SDP0	11	0b	Clears the mask for General Purpose Interrupt, related to SDP0 pin.
GPI_SDP1	12	0b	Clears the mask for General Purpose Interrupt, related to SDP1 pin.
GPI_SDP2	13	0b	Clears the mask for General Purpose Interrupt, related to SDP2 pin.
GPI_SDP3	14	0b	Clears the mask for General Purpose Interrupt, related to SDP3 pin.
PTRAP	15	0	Clears the mask for the Probe trap interrupt
Reserved	17:16	0b	Reserved.
MNG	18	0b	Clears the mask for Management Event Interrupt.
Reserved	19	0b	Reserved.
OMED	20	0b	Clears the mask for Other Media Energy Detected Interrupt.
Reserved	21	0b	Reserved
FER	22	0b	Clears the mask for the Fatal Error Interrupt.
NFER	23	0b	Clears the mask for the Non Fatal Error Interrupt.
CSRTO	24	0b	Clears the mask for the CSR access time out indication interrupt.
SCE	25	0b	Clears the mask for the Storm Control Event Interrupt.
Software WD	26	0b	Clears the mask for Software Watchdog Interrupt.
Reserved	27	0b	Reserved.
OUTSYNC	28	0b	Clears the mask for DMA Tx Out of Sync Interrupt.
Reserved	29	0b	
TCP timer	30	0b	Clears the mask for TCP timer interrupt.
Reserved	31	0000b	Reserved.

8.8.11 Interrupt Acknowledge Auto Mask Register - IAM (0x01510; R/W)

Field	Bit(s)	Initial Value	Description
IAM_VALUE	31:0	0b	Each time the *CTRL_EXT.IAME* bit is set, an ICR read or write has the side effect of writing the contents of this register to the IMC register.

8.8.12 Interrupt Throttle - EITR (0x01680 + 4*n [n = 0...24]; R/W)

Each EITR is responsible for an interrupt cause. The allocation of EITR-to-interrupt cause is through the IVAR registers.

Software uses this register to pace (or even out) the delivery of interrupts to the host processor. This register provides a guaranteed inter-interrupt delay between interrupts asserted by the 82576, regardless of network traffic conditions. To independently validate configuration settings, software can use the following algorithm to convert the inter-interrupt interval value to the common interrupts/sec. performance metric:

```
interrupts/sec = (1 * 10-6sec x interval)-1
```

A counter counts in units of 8*10-6 sec. After counting “interval “number of units, an interrupt is sent to the software. The above equation gives the number of interrupts per second. The equation below time in seconds between consecutive interrupts.

For example, if the interval is programmed to 125 (decimal), the 82576 guarantees the processor is not interrupted by it for 1 ms from the last interrupt. The maximum observable interrupt rate from the 82576 should never exceed 7813 interrupts/sec.

Inversely, inter-interrupt interval value can be calculated as:

$$\text{inter-interrupt interval} = (8 * 10^{-6}\text{sec x interrupt/sec})^{-1}$$

The optimal performance setting for this register is very system and configuration specific. An initial suggested range is 2 to 175 (2 - AF).

Note: Setting EITR to a non zero value can cause an interrupt cause Rx/Tx statistics miscount.

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0b	Reserved
Interval	14:2	0x0	Minimum inter-interrupt interval. The interval is specified in 1 µs increments. A zero disables interrupt throttling logic.
LLI_EN	15	0b	LLI moderation enable.

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0b	Reserved
LL Counter (RWS)	20:16	0x0	Reflects the current credits for that EITR for LL interrupts. If the CNT_INGR is not set this counter can be directly written by software at any time to alter the throttles performance
Moderation Counter (RWS)	30:21	0x0	Down counter, exposes only the 10 most significant bits of the real 12-bit counter. Loaded with Interval value whenever the associated interrupt is signaled. Counts down to 0 and stops. The associated interrupt is signaled whenever this counter is zero and an associated (via the Interrupt Select register) EICR bit is set. If the CNT_INGR is not set this counter can be directly written by software at any time to alter the throttles performance.
CNT_INGR (WO)	31	0b	When set the hardware does not override the counters fields (ITR counter and LLI credit counter), so they keep their previous value. Relevant for the current write only and is always read as zero

8.8.13 Interrupt Vector Allocation Registers - IVAR (0x1700 + 4*n [n=0...7]; RW)

These registers have two modes of operation:

1. In MSI-X mode these registers define the allocation of the different interrupt causes as defined in Table 7-43 to one of the MSI-X vectors. Each INT_Alloc[i] (i=0...31) field is a byte indexing an entry in the MSI-X Table Structure and MSI-X PBA Structure.
2. In non MSI-X mode these registers define the allocation of the Rx and Tx queues interrupt causes to one of the RxTxQ bits in the EICR. Each INT_Alloc[i] (i=0...31) field is a byte indexing the appropriate RxTxQ bit as defined in Table 7-42.

Field	Bit(s)	Initial Value	Description
INT_Alloc[0]	4:0	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-43. Valid values are 0 to 24 for MSI-X mode and 0 to 15 in non MSI-X mode.
Reserved	6:5	00b	Reserved
INT_Alloc_val[0]	7	0b	Valid bit for INT_Alloc[0]
INT_Alloc[1]	12:8	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-43. Valid values are 0 to 24 for MSI-X mode and 0 to 15 in non MSI-X mode.
Reserved	14:13	00b	Reserved
INT_Alloc_val[1]	15	0b	Valid bit for INT_Alloc[1]
INT_Alloc[2]	20:16	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-43. Valid values are 0 to 24 for MSI-X mode and 0 to 15 in non MSI-X mode.
Reserved	22:21	00b	Reserved
INT_Alloc_val[2]	23	0b	Valid bit for INT_Alloc[2]
INT_Alloc[3]	28:24	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-43. Valid values are 0 to 24 for MSI-X mode and 0 to 15 in non MSI-X mode.
Reserved	30:29	00b	Reserved
INT_Alloc_val[3]	31	0b	Valid bit for INT_Alloc[3]

Note: If invalid values are written to the INT_Alloc fields the result is unexpected.

DW	31 24	23 16	15 8	7 0
0	INT_ALLOC[3]	INT_ALLOC[2]	INT_ALLOC[1]	INT_ALLOC[0]
1			...	...
		. . .		
6	INT_ALLOC[31]	INT_ALLOC[30]	INT_ALLOC[29]	INT_ALLOC[28]

8.8.14 Interrupt Vector Allocation Registers - MISC IVAR_MISC (0x1740; RW)

This register is used only in MSI-X mode. This register defines the allocation of the “Other” and TCP timer interrupt causes to one of the MSI-X vectors.

Field	Bit(s)	Initial Value	Description
INT_Alloc[32]	4:0	0x0	Defines the MSI-X vector assigned to the TCP timer interrupt cause. Valid values are 0 to 24.
Reserved	6:5	00b	Reserved
INT_Alloc_val[32]	7	0b	Valid bit for INT_Alloc[32]
INT_Alloc[33]	12:8	0x0	Defines the MSI-X vector assigned to the “Other” interrupt cause. Valid values are 0 to 24.
Reserved	14:13	00b	Reserved
INT_Alloc_val[33]	15	0b	Valid bit for INT_Alloc[33]
Reserved	31:16	0x0	Reserved

8.8.15 General Purpose Interrupt Enable - GPIE (0x1514; RW)

Field	Bit(s)	Initial Value	Description
NSICR	0	0b	Non Selective Interrupt clear on read: When set, every read of ICR clears it. When this bit is cleared, an ICR read causes it to be cleared only if an actual interrupt was asserted or IMS = 0b.
Reserved	3:1	0x0	Reserved.
Multiple MSIX	4	0b	0 = on-MSIX, or MSI-X with single vector, IVAR map Rx/Tx causes to 16 EICR bits, but MSIX[0] is asserted for all. 1 = MSIX mode, IVAR maps Rx/Tx causes to 25 MSI-x vectors reflected in the first 25 bits of EICR.
Reserved	5	0b	Reserved
Reserved	6	0b	Reserved
LL Interval	11:7	0x0	Low latency credits increment rate. The interval is specified in 4 µs increments. A value of 0x0 disables moderation of LLI for all interrupt vectors.

Reserved	29:12	0x0	Reserved
EIAME	30	0b	Extended Interrupt Auto Mask enable: When set (usually in MSI-X mode); upon firing of an MSI-X message, bits set in EIAM associated with this message is cleared. Otherwise, EIAM is used only upon read or write of EICR/EICS registers.
PBA_ support	31	0b	PBA Support: When set, setting one of the extended interrupts masks via EIMS causes the PBA bit of the associated MSI-X vector to be cleared. Otherwise, the 82576 behaves in a way supporting legacy INT-x interrupts. **Note:** Should be cleared when working in INT-x or MSI mode and set in MSI-X mode.

8.9 MSI-X Table Register Descriptions

These registers are used to configure the MSI-X mechanism. The address and upper address registers sets the address for each of the vectors. The message register sets the data sent to the relevant address. The vector control registers are used to enable specific vectors.

The pending bit array register indicates which vectors have pending interrupts. The structure is listed in Table 8-20.

Table 8-20. MSI-X Table Structure

DWORD3	DWORD2	DWORD1	DWORD0		
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry 0	Base
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry 1	Base + 1*16
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry 2	Base + 2*16
...	...	...	...	...	
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry (N-1)	Base + (N-1) *16

Note: N = 25.

Table 8-21. MSI-X PBA Structure

63:0		
Pending Bits 0 through 63	QWORD0	Base
Pending Bits 64 through 127	QWORD1	Base+1*8
...	...	...
Pending Bits ((N-1) div 64)*64 through N-1	QWORD((N-1) div 64)	BASE + ((N-1) div 64)*8

Note: N = 25. As a result, only QWORD0 is implemented.

8.9.1 MSI–X Table Entry Lower Address - MSIXTADD (BAR3: 0x0000 + 0x10*n [n=0...24]; R/W)

Field	Bit(s)	Initial Value	Description
Message Address LSB (RO)	1:0	0x0	For proper DWORD alignment, software must always write 0b's to these two bits. Otherwise, the result is undefined.
Message Address	31:2	0x0	System-Specific Message Lower Address For MSI-X messages, the contents of this field from an MSI-X table entry specifies the lower portion of the DWORD-aligned address for the memory write transaction.

8.9.2 MSI–X Table Entry Upper Address - MSIXTUADD (BAR3: 0x0004 + 0x10*n [n=0...24]; R/W)

Field	Bit(s)	Initial Value	Description
Message Address	31:0	0x0	System-Specific Message Upper Address

8.9.3 MSI–X Table Entry Message - MSIXTMSG (BAR3: 0x0008 + 0x10*n [n=0...24]; R/W)

Field	Bit(s)	Initial Value	Description
Message Data	31:0	0x0	System-Specific Message Data For MSI-X messages, the contents of this field from an MSI-X table entry specifies the data written during the memory write transaction. In contrast to message data used for MSI messages, the low-order message data bits in MSI-X messages are not modified by the function.

8.9.4 MSI–X Table Entry Vector Control - MSIXTVCTRL (BAR3: 0x000C + 0x10*n [n=0...24]; R/W)

Field	Bit(s)	Initial Value	Description
Mask	0	1b	When this bit is set, the function is prohibited from sending a message using this MSI-X table entry. However, any other MSI-X table entries programmed with the same vector are still capable of sending an equivalent message unless they are also masked.
Reserved	31:1	0x0	Reserved

8.9.5 MSIXPBA Bit Description – MSIXPBA (BAR3: 0x02000; RO)

Field	Bit(s)	Initial Value	Description
Pending Bits	24:0	0x0	For each pending bit that is set, the function has a pending message for the associated MSI-X Table entry. Pending bits that have no associated MSI-X table entry are reserved.
Reserved	31:25	0x0	Reserved

8.9.6 MSI-X PBA Clear – PBACL (0x05B68; R/W1C)

Field	Bit(s)	Initial Value	Description
PENBIT	24:0	0x0	MSI-X Pending bits Clear Writing a 1b to any bit clears the corresponding MSIXPBA bit; writing a 0b has no effect. Reading this register returns the PBA vector.
Reserved	31:25	0x0	Reserved

8.10 Receive Register Descriptions

8.10.1 Receive Control Register - RCTL (0x00100; R/W)

This register controls all the 82576 receiver functions.

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write to 0b for future compatibility.
RXEN	1	0b	Receiver Enable The receiver is enabled when this bit is set to 1b. Writing this bit to 0b stops reception after receipt of any in progress packet. All subsequent packets are then immediately dropped until this bit is set to 1b.
SBP	2	0b	Store Bad Packets 0b = do not store. 1b = store bad packets. This bit controls the MAC receive behavior. A packet is required to pass the address (or normal) filtering before the SBP bit becomes effective. If SBP = 0b, then all packets with layer 1 or 2 errors are rejected. The appropriate statistic would be incremented. If SBP = 1b, then these packets are received (and transferred to host memory). The receive descriptor error field (RDESC.ERRORS) should have the corresponding bit(s) set to signal the software device driver that the packet is erred. In some operating systems the software device driver passes this information to the protocol stack. In either case, if a packet only has layer 3+ errors, such as IP or TCP checksum errors, and passes other filters, the packet is always received (layer 3+ errors are not used as a packet filter). **Note:** symbol errors before the SFD are ignored. Any packet must have a valid SFD (RX_DV with no RX_ER in 10/100/1000BASE-T mode) in order to be recognized by the 82576 (even bad packets). Also, erred packets are not routed to the MNG even if this bit is set.

Field	Bit(s)	Initial Value	Description
UPE	3	0b	Unicast Promiscuous Enabled 0b = Disabled. 1b = Enabled.
MPE	4	0b	Multicast Promiscuous Enabled 0b = Disabled. 1b = Enabled.
LPE	5	0b	Long Packet Reception Enable 0b = Disabled. 1b = Enabled. LPE controls whether long packet reception is permitted. Hardware discards long packets if LPE is 0b. A long packet is one longer than 1522 bytes. If LPE is 1b, the maximum packet size that the 82576 can receive is 9.5Kbytes.
LBM	7:6	00b	Loopback mode. Controls the loopback mode of the 82576. 00b = Normal operation (or PHY loopback in 10/100/1000BASE-T mode). 01b = MAC loopback (test mode). 10b = Undefined. 11b = Reserved. When using the internal PHY, LBM should remain set to 00b and the PHY instead configured for loopback through the MDIO interface. **Note:** PHY devices require programming for loopback operation using MDIO accesses.
Reserved	9:8	00b	Reserved
Reserved	11:10	00b	Reserved Set to 0b for compatibility.
MO	13:12	00b	Multicast Offset Determines which bits of the incoming multicast address are used in looking up the 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
BAM	15	0b	Broadcast Accept Mode. 0b = Ignore broadcast (unless it matches through exact or imperfect filters). 1b = Accept broadcast packets.
BSIZE	17:16	00b	Receive Buffer Size BSIZE controls the size of the receive buffers and permits software to trade-off descriptor performance versus required storage space. Buffers that are 2048 bytes require only one descriptor per receive packet maximizing descriptor efficiency. 00b = 2048 Bytes. 01b = 1024 Bytes. 10b = 512 Bytes. 11b = 256 Bytes. **Note:** BSIZE is not modified when RXEN is set to 1b. Set RXEN =0 when modifying the buffer size (changing these bits).

Field	Bit(s)	Initial Value	Description
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 [20:18] control the VLAN filter table. The first determines whether the table participates in the packet acceptance criteria. The next two are used to decide whether the CFI bit found in the 802.1Q packet 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 0b = 802.1Q packets with CFI equal to this field are accepted. 1b = 802.1Q packet is discarded.
PSP	21	0b	Pad Small Receive packets. If this field is set, SECRC should be set also.
DPF	22	0b	Discard Pause Frames with Station MAC Address Controls whether pause frames directly addressed to this station are forwarded to the host. 0b = incoming pause frames with station MAC address are forwarded to the host. 1b = incoming pause frames with station MAC address are discarded. **Note:** Pause frames with other MAC addresses (multicast address) are always discarded unless the specific address is added to the accepted MAC addresses (either multicast or unicast).
PMCF	23	0b	Pass MAC Control Frames Filters out unrecognized pause and other control frames. 0b = Pass/forward pause frames. 1b = Filter pause frames (default). 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 0x0001. If PMCF = 1b then frames meeting this criteria are transferred to host memory.
Reserved	25:24	0b	Reserved Should be written with 0b to ensure future compatibility.
SECRC	26	0b	Strip Ethernet CRC from incoming packet Causes the CRC to be stripped from all packets. 0b = Does not strip CRC 1b = Strips CRC. This bit controls whether the hardware strips the Ethernet CRC from the received packet. This stripping occurs prior to any checksum calculations. The stripped CRC is not transferred to host memory and is not included in the length reported in the descriptor.
Reserved	31:27	0x0	Reserved Should be written with 0b to ensure future compatibility.

8.10.2 Split and Replication Receive Control - SRRCTL (0x0C00C + 0x40*n [n=0...15]; R/W)

Field	Bit(s)	Initial Value	Description
BSIZEPACKET	6:0	0x0	Receive Buffer Size for Packet Buffer The value is in 1 KB resolution. Valid values can be from 1 KB to 127 KB. Default buffer size is 0 KB. If this field is equal 0b, then RCTL.BSIZE determines the packet buffer size.
Reserved	7	0x0	Reserved
BSIZEHEADER	11:8	0x4	Receive Buffer Size for Header Buffer The value is in 64 bytes resolution. Valid value scan be from 64 bytes to 1024 bytes. Default buffer size is 256 bytes. This field must be greater than 0 if the value of DESCTYPE is greater or equal to 2.
Reserved	13:12	00b	Reserved Must be set to 00b.
Reserved	19:14	0x0	Reserved
RDMTS	24:20	0x0	Receive Descriptor Minimum Threshold Size A low latency interrupt (LLI) associated with this queue is asserted whenever the number of free descriptors becomes equal to RDMTS multiplied by 16.
DESCTYPE	27:25	000b	Defines the descriptor in Rx 000b = Legacy. 001b = Advanced descriptor one buffer. 010b = Advanced descriptor header splitting. 011b = Advanced descriptor header replication - replicate always. 100b = Advanced descriptor header replication large packet only (larger than header buffer size). 101b = Reserved. 111b = Reserved.
Reserved	30:28	0x0	Reserved
Drop_En	31	0b/1b	Drop Enabled If set, packets received to the queue when no descriptors are available to store them are dropped. The packet is dropped only if there are not enough free descriptors in the host descriptor ring to store the packet. If there are enough descriptors in the host, but they are not yet fetched by the 82576, then the packet is not dropped and there are no release of packets until the descriptors are fetched. Default is 0b for queue 0 and 1b for the other queues.

8.10.3 Packet Split Receive Type - PSRTYPE (0x05480 + 4*n [n=0...7]; R/W)

This register enables or disables each type of header that needs to be split. Each register controls the behavior of 2 queues.

- Packet Split Receive Type Register (queue 0-1) - PSRTYPE0 (0x05480)
- Packet Split Receive Type Register (queue 2-3) - PSRTYPE1 (0x05484)
- Packet Split Receive Type Register (queue 4-5) - PSRTYPE2 (0x05488)
- Packet Split Receive Type Register (queue 6-7) - PSRTYPE3 (0x0548C)
- Packet Split Receive Type Register (queue 8-9) - PSRTYPE4 (0x05490)
- Packet Split Receive Type Register (queue 10-11) - PSRTYPE5 (0x05494)

- Packet Split Receive Type Register (queue 12-13) - PSRTYPE6 (0x05498)
- Packet Split Receive Type Register (queue 14-15) - PSRTYPE7 (0x0549C)

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved
PSR_type1	1	1b	Header includes MAC, (VLAN/SNAP) IPv4 only
PSR_type2	2	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP only
PSR_type3	3	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP only
PSR_type4	4	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6 only
PSR_type5	5	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP only
PSR_type6	6	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP only
PSR_type7	7	1b	Header includes MAC, (VLAN/SNAP) IPv6 only
PSR_type8	8	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP only
PSR_type9	9	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP only
Reserved	10	1b	Reserved
PSR_type11	11	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP, NFS only
PSR_type12	12	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP, NFS only
Reserved	13	1b	Reserved
PSR_type14	14	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP, NFS only
PSR_type15	15	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP, NFS only
Reserved	16	1b	Reserved
PSR_type17	17	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP, NFS only
PSR_type18	18	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP, NFS only
Reserved	31:19	0x0	Reserved

8.10.4 Replicated Packet Split Receive Type - RPLPSRTYPE (0x054C0; R/W)

This register enables or disables each type of header that needs to be split. This register controls the behavior of replicated packets.

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved
PSR_type1	1	1b	Header includes MAC, (VLAN/SNAP) IPv4 only
PSR_type2	2	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP only
PSR_type3	3	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP only
PSR_type4	4	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6 only
PSR_type5	5	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP only
PSR_type6	6	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP only
PSR_type7	7	1b	Header includes MAC, (VLAN/SNAP) IPv6 only
PSR_type8	8	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP only
PSR_type9	9	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP only
Reserved	10	1b	Reserved
PSR_type11	11	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP, NFS only

Field	Bit(s)	Initial Value	Description
PSR_type12	12	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP, NFS only
Reserved	13	1b	Reserved
PSR_type14	14	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP, NFS only
PSR_type15	15	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP, NFS only
Reserved	16	1b	Reserved
PSR_type17	17	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP, NFS only
PSR_type18	18	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP, NFS only
Reserved	31:19	0x0	Reserved

8.10.5 Receive Descriptor Base Address Low - RDBAL (0x0C000 + 0x40*n [n=0...15]; R/W)

This register contains the lower bits of the 64-bit descriptor base address. The lower four bits are always ignored. The Receive Descriptor Base Address must point to a 128 byte-aligned block of data.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2800, 0x2900, 0x2A00 & 0x2B00 respectively.

Field	Bit(s)	Initial Value	Description
0	6:0	0x0	Ignored on writes. Returns 0b on reads.
RDBAL	31:7	X	Receive Descriptor Base Address Low

8.10.6 Receive Descriptor Base Address High - RDBAH (0x0C004 + 0x40*n [n=0...15]; R/W)

This register contains the upper 32 bits of the 64-bit descriptor base address.

Field	Bit(s)	Initial Value	Description
RDBAH	31:0	X	Receive Descriptor Base Address [63:32]

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2804, 0x2904, 0x2A04 & 0x2B04 respectively.

8.10.7 Receive Descriptor Ring Length - RDLEN (0x0C008 + 0x40*n [n=0...15]; R/W)

This register sets the number of bytes allocated for descriptors in the circular descriptor buffer. It must be 128-byte aligned.

Field	Bit(s)	Initial Value	Description
LEN	19:0	0x0	Descriptor Ring Length (in bytes). Bits 6:0 must be set to zero. Bits 3:0 always reads as zero.
Reserved	31:20	0x0	Reserved Reads as 0b. Should be written to 0b for future compatibility.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2808, 0x2908, 0x2A08 & 0x2B08 respectively.

8.10.8 Receive Descriptor Head - RDH (0x0C010 + 0x40*n [n=0...15]; R/W)

The value in this register might point to descriptors that are still not in host memory. As a result, the host cannot rely on this value in order to determine which descriptor to process.

Field	Bit(s)	Initial Value	Description
RDH	15:0	0x0	Receive Descriptor Head
Reserved	31:16	0x0	Reserved Should be written to 0b.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2810, 0x2910, 0x2A10 & 0x2B10 respectively.

8.10.9 Receive Descriptor Tail - RDT (0x0C018 + 0x40*n [n=0...15]; 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.

Note: Writing the RDT register while the corresponding queue is disabled is ignored by the 82576.

In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2818, 0x2918, 0x2A18& 0x2B18 respectively.

Field	Bit(s)	Initial Value	Description
RDT	15:0	0x0	Receive Descriptor Tail
Reserved	31:16	0x0	Reserved Reads as 0b. Should be written to 0b for future compatibility.

8.10.10 Receive Descriptor Control - RXDCTL (0x0C028 + 0x40*n [n=0...15]; 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 are in units of descriptors (each descriptor is 16 bytes).

Field	Bit(s)	Initial Value	Description
PTHRESH	4:0	0x0	Prefetch Threshold PTHRESH is used to control when a prefetch of descriptors is considered. This threshold refers to the number of valid, unprocessed receive descriptors the 82576 has in its on-chip buffer. If this number drops below PTHRESH, the algorithm considers pre-fetching descriptors from host memory. This fetch does not happen unless there are at least HTHRESH valid descriptors in host memory to fetch. **Note:** HTHRESH should be given a non zero value each time PTHRESH is used. Possible values for this field are 0 to 16.
Reserved	7:5	0x0	Reserved
HTHRESH	12:8	0x0	Host Threshold Possible values for this field are 0 to 16.
Reserved	15:13	0x0	Reserved
WTHRESH	20:16	0x01	Write-Back Threshold WTHRESH controls the write-back of processed receive descriptors. This threshold refers to the number of receive descriptors in the on-chip buffer that are ready to be written back to host memory. In the absence of external events (explicit flushes), the write-back occurs only after at least WTHRESH descriptors are available for write-back. Possible values for this field are 0 to 31: **Note:** Since the default value for write-back threshold is 1b, the descriptors are normally written back as soon as one cache line is available. WTHRESH must contain a non-zero value to take advantage of the write-back bursting capabilities of the 82576.
Reserved	24:21	0x0	Reserved
ENABLE	25	1b/0b	Receive Queue Enable When set, the *Enable* bit enables the operation of the specific receive queue. 1b =Enables queue. 0b =Disables queue. Default value for Q0 is 1b. Default value for Q15:1 is 0b. After a VF FLR to VF0, Q0 is also reset to zero. Setting this bit initializes all internal registers of the specific queue. Until then, the state of the queue is kept and can be used for debug purposes. When disabling a queue, this bit is cleared only after all activity in the queue has stopped. Note: This bit is valid only if the queue is actually enabled, thus if RCTL.RXEN is cleared, this bit remains zero.
SWFLUSH (WC)	26	0b	Receive Software Flush Enables software to trigger receive descriptor write-back flushing, independently of other conditions. This bit is cleared by hardware.
Reserved	27	0x00	Reserved
Reserved	28	0	Reserved
Reserved	29	0	Reserved
Reserved	31:30	0x00	Reserved

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2828, 0x2928, 0x2A28 & 0x2B28 respectively.

8.10.11 Receive Queue Drop Packet Count - RQDPC (0xC030 + 0x40*n [n=0...15]; RC)

Field	Bit(s)	Initial Value	Description
RQDPC	11:0	0x0	Receive Queue drop packet count - counts the number of packets dropped by a queue due to lack of descriptors available.
Reserved	31:12	0x0	REserved

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2830, 0x2930, 0x2A30 & 0x2B30 respectively.

Packets dropped due to the queue being disabled may not be counted by this register.

8.10.12 DMA RX Max Outstanding Data - DRXMXOD (0x2540; RW)

This register limits the total number of data bytes that might be in the write pipe to the host memory. This allows received low latency packets to be serviced in a timely manner, as this limits the amount of data to be processed before the low latency packet is handled.

Field	Bit(s)	Initial Value	Description
Max_bytes_num_req	11:0	0x10	Max allowed number of bytes requests. The maximum size of the data in the write pipe (resolution is 256 bytes). If the total size is higher than the amount in the field no arbitration is done and no new packet is sent.
Reserved	31:12	0x0	Reserved

8.10.13 Receive Checksum Control - RXCSUM (0x05000; R/W)

The Receive Checksum Control register controls the receive checksum off loading features of the 82576. The 82576 supports the off loading of three receive checksum calculations: the Packet Checksum, the IP Header Checksum, and the TCP/UDP Checksum.

Note: This register should only be initialized (written) when the receiver is not enabled (only write this register when RCTL.RXEN = 0b)

Field	Bit(s)	Initial Value	Description
PCSS	7:0	0x0	Packet Checksum Start Controls the packet checksum calculation. The packet checksum shares the same location as the RSS field and is reported in the receive descriptor when the RXCSUM.PCSD bit is cleared. If RXCSUM.IPPCSE is cleared (the default value), the checksum calculation that is reported in the Rx Packet checksum field is the unadjusted 16-bit ones complement of the packet. The packet checksum starts from the byte indicated by RXCSUM.PCSS (0b corresponds to the first byte of the packet), after VLAN stripping if enabled by the CTRL.VME. 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 L4 checksum stored in the packet checksum. The partial checksum in the descriptor is aimed to accelerate checksum calculation of fragmented UDP packets. **Note:** The PCSS value should not exceed a pointer to the IP header start. If exceeded, the IP header checksum or TCP/UDP checksum is not calculated correctly.
IPOFLD	8	1b	IP Checksum Off-load Enable RXCSUM.IPOFLD is used to enable the IP Checksum off-loading feature. If RXCSUM.IPOFLD is set to 1b, the 82576 calculates the IP checksum and indicates a pass/fail indication to software via the IP Checksum Error bit (IPE) in the *Error* field of the receive descriptor. Similarly, if RXCSUM.TUOFLD is set to 1b, the 82576 calculates the TCP or UDP checksum and indicates a pass/fail indication to software via the TCP/UDP Checksum Error bit (L4E). Similarly, if RFCTL.IPv6_DIS and RFCTL.IP6Xsum_DIS are cleared to 0b and RXCSUM.TUOFLD is set to 1b, the 82576 calculates the TCP or UDP checksum for IPv6 packets. It then indicates a pass/fail condition in the TCP/UDP *Checksum Error* bit (RDESC.L4E). This applies to checksum off loading only. Supported frame types: Ethernet II Ethernet SNAP
TUOFLD	9	1b	TCP/UDP Checksum Off-load Enable
Reserved	10	0b	Reserved
CRCOFL	11	0b	CRC32 Offload Enable Enables the CRC32 checksum off-loading feature. If RXCSUM.CRCOFL is set to 1b, the 82576 calculates the CRC32 checksum and indicates a pass/fail indication to software via the CRC32 *Checksum Valid* bit (CRCV) in the *Extended Status* field of the receive descriptor. In non I/OAT, this bit is read only as 0b.

IPPCSE	12	0b	IP Payload Checksum Enable See PCSS description.
PCSD	13	0b	Packet Checksum Disable The packet checksum and IP identification fields are mutually exclusive with the RSS hash. Only one of the two options is reported in the Rx descriptor. RXCSUM.PCSD Legacy Rx Descriptor (SRRCTL.DESCTYPE != 000b): 0b (checksum enable) - Packet checksum is reported in the Rx descriptor. 1b (checksum disable) - Not supported. RXCSUM.PCSD Extended or Header Split Rx Descriptor (SRRCTL.DESCTYPE != 000b): 0b (checksum enable) - checksum and IP identification are reported in the Rx descriptor. 1b (checksum disable) - RSS Hash value is reported in the Rx descriptor.
Reserved	31:14	0x0	Reserved

8.10.14 Receive Long Packet Maximum Length - RLPML (0x5004; R/W)

Field	Bit(s)	Initial Value	Description
RLPML	13:0	0x2600	Maximum allowed long packet length. This length is the global length of the packet including all the potential headers of suffixes in the packet.
Reserved	31:14	0x0	Reserved

8.10.15 Receive Filter Control Register - RFCTL (0x05008; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	5:0	1b	Reserved
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.
IPv6_DIS	10	0b	IPv6 Disable Disables IPv6 packet filtering. Any received IPv6 packet is parsed only as an L2 packet.
IPv6XSUM_DIS	11	0b	IPv6 XSUM Disable Disables XSUM on IPv6 packets.
Reserved	12	0b	Reserved
Reserved	13	0b	Reserved (was ACK accelerate disable & ACK data Disable).
IPFRSP_DIS	14	0b	IP Fragment Split Disable When this bit is set, the header of IP fragmented packets are not set.
Reserved	15	0b	Reserved

Reserved	17:16	00b	Reserved Must be set to 00b.
LEF	18	0b	Forward Length Error Packet 0b = packet with length error are dropped. 1b = packets with length error are forwarded to the host.
SYNQFP	19	0b	Defines the priority between SYNQF & 5 tuples filter 0b = 5-tuple filter priority 1b = SYN filter priority.
Reserved	31:20	0x08	Reserved Should be written with 0b to ensure future capability.

8.10.16 Multicast Table Array - MTA (0x05200 + 4*n [n=0...127]; R/W)

There is one register per 32 bits of the Multicast Address Table for a total of 128 registers. Software must mask to the desired bit on reads and supply a 32-bit word on writes. The first bit of the address used to access the table is set according to the RX_CTRL.MO field.

Note: All accesses to this table must be 32 bit.

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Word wide bit vector specifying 32 bits in the multicast address filter table.

Figure 8-1 shows the multicast lookup algorithm. The destination address shown represents the internally stored ordering of the received DA. Note that bit 0 indicated in this diagram is the first on the wire.

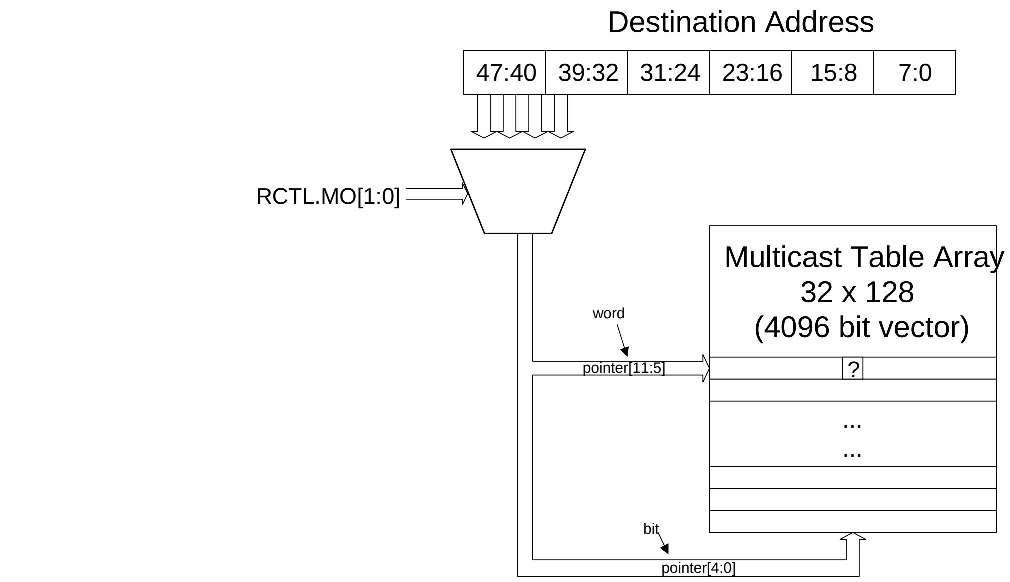


Figure 8-1. Multicast Table Array

8.10.17 Receive Address Low - RAL (0x05400 + 8*n [n=0...15]; 0x054E00 + 8*n [n=0...7]; R/W)

While “n” is the exact unicast/multicast address entry and it is equal to 0,1,...15.

These registers contain the lower bits of the 48 bit Ethernet address. All 32 bits are valid.

These registers are reset by a software reset or platform reset. If an EEPROM is present, the first register (RAL0) is loaded from the EEPROM after a software or platform reset.

Note: The *RAL* field should be written in network order.

Field	Bit(s)	Initial Value	Description
RAL	31:0	X	Receive address low Contains the lower 32-bit of the 48-bit Ethernet address.

8.10.18 Receive Address High - RAH (0x05404 + 8*n [n=0...15]; 0x054E04 + 8*n [n=0...7]; 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 and is cleared by a master reset.

ASEL enables the 82576 to perform special filtering on receive packets.

After reset, if an EEPROM is present, the first register (Receive Address Register 0) is loaded from the *IA* field in the EEPROM with its *Address Select* field set to 00b and its *Address Valid* field set to 1b. If no EEPROM is present, the *Address Valid* field is set to 0b and the *Address Valid* field for all of the other registers is set to 0b.

Note: The *RAH* field should be written in network order.

The first receive address register (RAH0) is also used for exact match pause frame checking (DA matches the first register). As a result, RAH0 should always be used to store the individual Ethernet MAC address of the 82576.

Field	Bit(s)	Initial Value	Description
RAH	15:0	X	Receive address High Contains the upper 16 bits of the 48-bit Ethernet 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. This mode should not be used in virtualization mode. 10b = Reserved 11b = Reserved
POOLSEL	25:18	0x0	Pool Select In virtualization modes (MRQC.Multiple Receive Queues Enable = 011b - 101b) indicates which Pool should get the packets matching this MAC address. This field is a bit map (bit per VM) where more than one bit can be set according to the limitations defined in Section 7.9.3.4. If all the bits are zero, this address is used only for L2 filtering and is not used as part of the queueing decision.
Reserved	30:26	0b	Reserved Reads as 0b. Ignored on writes.
AV	31		Address Valid Cleared after master reset. If an EEPROM is present, the *Address Valid* field of the Receive Address Register 0 is set to 1b after a software or PCI reset or EEPROM read. In entries 0-15 this bit is cleared by master reset.

8.10.19 VLAN Filter Table Array - VFTA (0x05600 + 4*n [n=0...127]; R/W)

There is one register per 32 bits of the VLAN Filter Table. 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.

Note: All 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 8.10.16 for a block diagram of the algorithm. If VLANs are not used, there is no need to initialize the VFTA.

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Double-word wide bit vector specifying 32 bits in the VLAN Filter table.

8.10.20 Multiple Receive Queues Command Register - MRQC (0x05818; R/W)

Field	Bit(s)	Initial Value	Description
Multiple Receive Queues Enable	2:0	0x0	Multiple Receive Queues Enable Enables support for Multiple Receive Queues and defines the mechanism that controls queue allocation. 000b = Multiple receive queues are disabled. 001b = Reserved. 010b = Multiple receive queues as defined by RSS for sixteen queues[1]. 011b = Multiple receive queues as defined by Next Generation VMDq based on packet destination MAC address. In this case, all the packets are forwarded to the default queue of each pool. 100b = Multiple receive queues as defined by Next Generation VMDq based on packet destination MAC address. 101b = Multiple receive queues as defined by Next Generation VMDq based on packet destination MAC address and RSS[1]. 110b = Multiple receive queues as defined by RSS[1]. 111b = Reserved. If VT is not supported, the only allowed values for this field are 000b, 001b, 010b and 110b. Writing any other value is ignored. The only allowed values for this field are 000b, 001b, 011b, and 101b. Writing any other value is ignored.
Def_Q	6:3	0x0	Defines the default queue in non Next Generation VMDq modes. If Multiple Receive Queues Enable = 000b: Defines the destination of all packets 001b: bits 5:3 defines the LSB of the queue number 010b: Defines the destination of all packets not forwarded by RSS 010b - 101b: This field is ignored. 110b: bits 5:3 defines the LSB of the queue number of all packets not forwarded by RSS.
Reserved	15:7	0x0	Reserved.

RSS Field Enable	31:16	0x0	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
			Bit[21] = Enable TCPIPv6 hash function
			Bit[22] = Enable UDPIPv4
			Bit[23] = Enable UDPIPv6
			Bit[24] = Enable UDPIPv6Ext
			ReservedBits[31:26] - Reserved Zero

[1] Note that the *RXCSUM.PCSD* bit should be set to enable reception of the RSS hash value in the receive descriptor.

Note: MRQC_EN is used to enable/disable RSS hashing and also to enable multiple receive queues. Disabling this feature is not recommended. Model usage is to reset the 82576 after disabling the RSS.

8.10.21 RSS Random Key Register - RSSRK (0x05C80 + 4*n [n=0...9]; R/W)

Field	Bit(s)	Initial Value	Description
K0	7:0	0x0	Byte n*4 of the RSS random key (n=0,1,...9).
K1	15:8	0x0	Byte n*4+1 of the RSS random key (n=0,1,...9).
K2	23:16	0x0	Byte n*4+2 of the RSS random key (n=0,1,...9).
K3	31:24	0x0	Byte n*4+3 of the RSS random key (n=0,1,...9).

The RSS Random Key Register stores a 40 byte key used by the RSS hash function.

31 24	23 16	15 8	7 0
K[3]	K[2]	K[1]	K[0]
. . .	. . .	. . .	. . .
K[39]	. . .	. . .	K[36]

8.10.22 Redirection Table - RETA (0x05C00 + 4*n [n=0...31]; R/W)

The redirection table is a 128-entry table with each entry being eight bits wide. Only one to four bits of each entry are used to store the queue index. The table is configured through the following R/W registers.

Field	Bit(s)	Initial Value	Description
Entry 0	7:0	0x0	Determines the tag value and physical queue for index 4*n+0 (n=0...31).
Entry 1	15:8	0x0	Determines the tag value and physical queue for index 4*n+1 (n=0...31).
Entry 2	23:16	0x0	Determines the tag value and physical queue for index 4*n+2 (n=0...31).
Entry 3	31:24	0x0	Determines the tag value and physical queue for index 4*n+3 (n=0...31).

31 24	23 16	15 8	7 0
Tag 3	Tag 2	Tag 1	Tag 0
. . .	. . .	. . .	. . .
Tag 127	. . .	. . .	. . .

Each entry (byte) of the redirection table contains the following:

7:4	3:0
Reserved	Queue index

- Bits [7:4] - Reserved

- Bits [3:0] - Queue index for all pools or in regular RSS. In RSS mode, all bits are used. In Next Generation VMDq + RSS mode only bit 0 is used

The contents of the redirection table are not defined following reset of the Memory Configuration Registers. System software must initialize the Table prior to enabling multiple receive queues. It can also 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 takes effect on a specific packet boundary.

Note: In case the operating system provides a redirection Table whose size is smaller than 128 bytes, the software usually replicates the operating system-provided redirection table to span the whole 128 bytes of the hardware's redirection table.

8.11 Filtering Register Descriptions

8.11.1 Immediate Interrupt Rx - IMIR (0x05A80 + 4*n [n=0...7]; R/W)

This register defines the filtering that corrects which packet triggers low latency interrupt. Another register includes a size threshold and a control bits bitmap to trigger an immediate interrupt.

Note: The *Port* field should be written in network order.

If one of the actions for this filter is set, then at least one of the PORT_BP, Size_BP, one of Mask bit or CtrlBit_BP bits should be cleared.

Field	Bit(s)	Initial Value	Description
Destination Port	15:0	0x0	Destination TCP port This field is compared with the Destination TCP port in incoming packets.
Immediate Interrupt	16	0b	Enables issuing an immediate interrupt when the following conditions are met: • The 5-tuple filter associated with this register matches • The Length filter associated with this filter matches • The TCP flags filter associated with this filter matches
PORT_BP	17	X	Port Bypass When set to 1b, the TCP port check is bypassed and only other conditions are checked. When set to 0b, the TCP port is checked to fit the port field.
Reserved	28:18	0x0	Reserved
Filter Priority	31:29	000b	Defines the priority of the filter assuming two filters matches. If two filter of the same priority matches the incoming packet, any of the highest priority filters can be chosen.

8.11.2 Immediate Interrupt Rx Ext. - IMIREXT (0x05AA0 + 4*n [n=0...7]; R/W)

Field	Bit(s)	Initial Value	Description
Size_Thresh	11:0	X	Size Threshold These 12 bits define a size threshold; a packet with a length below this threshold triggers an interrupt. Enabled by Size_Thresh_en.
Size_BP	12	X	Size Bypass When 1b, the size check is bypassed. When 0b, the size check is performed.
CtrlBit	18:13	X	Control Bit When a bit in this field equals 1b, an interrupt is immediately issued after receiving a packet with the corresponding TCP control bits turned on. Bit 13 (URG): Urgent pointer field significant Bit 14 (ACK): Acknowledgment field Bit 15 (PSH): Push function Bit 16 (RST): Reset the connection Bit 17 (SYN): Synchronize sequence numbers Bit 18 (FIN): No more data from sender
CtrlBit_BP	19	X	Control Bits Bypass When set to 1b, the control bits check is bypassed. When set to 0b, the control bits check is performed.
Reserved	31:20	0x0	Reserved

Note: The size used for this comparison is the size of the packet as forwarded to the host and does not include any of the fields stripped by the MAC (VLAN or CRC). As a result, setting the RCTL.SECRC & CTRL.VME bits should be taken into account while calculating the size threshold.

The value of the IMIR and IMIREXT registers after reset is unknown (apart from the IMIR.PORT_IM_EN bit which is guaranteed to be cleared). Therefore, both registers should be programmed before IMIR.PORT_IM_EN is set for a given flow.

8.11.3 Source Address Queue Filter - SAQF (0x5980 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Source Address	31:0	0x0	IP source address, part of the 5-tuple queue filters

8.11.4 Destination Address Queue Filter - DAQF (0x59A0 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Destination Address	31:0	0x0	IP destination address, part of the 5-tuple queue filters

8.11.5 Source Port Queue Filter - SPQF (0x59C0 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Source Port	15:0	0x0	TCP/UDP source port, part of the 5-tuple queue filters
Reserved	31:16	0x0	Reserved

8.11.6 5-tuple Queue Filter - FTQF (0x59E0 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Protocol	7:0	0x0	IP L4 protocol, part of the 5-tuple queue filters.
Queue Enable	8	0b	When set, enables filtering of Rx packets by the 5-tuple defined in this filter to the queue indicated in this register.
VF	11:9	0x0	The VF Index of the VF associated with this filter.
Reserved	14:12	0	Reserved (for extension of number of VFs).
VF Mask	15	1b (for legacy reasons)	Mask bit for the VF field. When set to 1b, the VF field is not compared as part of the 5-tuple filter. Software can clear (activate) the Pool Mask bit only when operating in virtualization mode.
Rx Queue	25:16	0x0	Identifies the Rx queue associated with this 5-tuple filter. If the VF Mask bit is set, the queue number is used as an offset to the VM list and do not override it.
Reserved	26	0b	Reserved
1588 time stamp	27	0b	When set, packets that match this filter are time stamped according to the IEEE 1588 specification.
Mask	31:28	0xF (for legacy reasons)	Mask bits for the 5-tuple fields (the mask bit for destination port is in the IMIR register for legacy reasons). The corresponding field participates in the match if the bit below is cleared: Bit 28 - Mask protocol comparison Bit 29 - Mask source address comparison Bit 30 - Mask destination address comparison Bit 31 - Mask source port comparison

8.11.7 Immediate Interrupt Rx VLAN Priority - IMIRVP (0x05AC0; R/W)

Field	Bit(s)	Initial Value	Description
Vlan_Pri	2:0	000b	VLAN Priority This field includes the VLAN priority threshold. When Vlan_pri_en is set to 1b, then an incoming packet with a VLAN tag with a priority field equal or higher to VlanPri triggers an immediate interrupt, regardless of the EITR moderation.
Vlan_pri_en	3	0b	VLAN Priority Enable When set to 1b, an incoming packet with VLAN tag with a priority equal or higher to Vlan_Pri triggers an immediate interrupt, regardless of the EITR moderation. When set to 0b, the interrupt is moderated by EITR.
Reserved	31:4	0x0	Reserved

8.11.8 SYN Packet Queue Filter - SYNQF (0x55FC; RW)

Field	Bit(s)	Initial Value	Description
Queue Enable	0	0b	When set, enables forwarding of Rx packets to the queue indicated in this register.
Rx Queue	4:1	0x0	Identifies an Rx queue associated with SYN packets.
Reserved	31:5	0x0	Reserved

8.11.9 EType Queue Filter - ETQF (0x5CB0 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
EType	15:0	0x0	Identifies the protocol running on top of IEEE 802. Used to forward Rx packets containing this EType to a specific Rx queue.
Rx Queue	19:16	0x0	Identifies the Rx queue associated with this EType.
Reserved	25:20	0x0	Reserved
Filter enable	26	0b	When set, this filter is valid. Any of the actions controlled by the following fields are gated by this field.
Reserved	28:27	0b	Reserved
BCN frame	28	0x0	When set, packets with this EType are parsed according to the BCN specification.
Immediate Interrupt	29	0x0	When set, packets that match this filter generate an immediate interrupt.
1588 time stamp	30	0b	When set, packets with this EType are time stamped according to the IEEE 1588 specification.
Queue Enable	31	0b	When set, enables filtering of Rx packets by the EType defined in this register to the queue indicated in this register.

8.12 Transmit Register Descriptions

8.12.1 Transmit Control Register - TCTL (0x00400; R/W)

This register controls all transmit functions for the 82576.

Software can choose to abort packet transmission in less than the Ethernet mandated 16 collisions. For this reason, hardware provides CT.

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 82576. Software should not write a 1b to this bit while the 82576 is configured for half-duplex operation.

RTLC configures the 82576 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.

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write as 0b for future compatibility.
EN	1	0b	Transmit Enable The transmitter is enabled when this bit is set to 1b. Writing 0b to this bit stops transmission after any in progress packets are sent. Data remains in the transmit FIFO until the device is re-enabled. Software should combine this operation with reset if the packets in the TX FIFO should be flushed.
Reserved	2	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.
PSP	3	1b	Pad Short Packets 0b = Do not pad. 1b = Pad. Padding makes the packet 64 bytes long. This is not the same as the minimum collision distance. If padding of short packets is allowed, the total length of a packet not including FCS should be not less than 17 bytes.
CT	11:4	0xF	Collision Threshold This determines the number of attempts at retransmission prior to giving up on the packet (not including the first transmission attempt). While this can be varied, it should be set to a value of 15 in order to comply with the IEEE specification requiring a total of 16 attempts. The Ethernet back-off algorithm is implemented and clamps to the maximum number of slot-times after 10 retries. This field only has meaning when in half-duplex operation.
BST	21:12	0x40	Back-Off Slot Time This value determines the back-off slot time value in byte time.
SWXOFF	22	0b	Software XOFF Transmission When set to 1b, the 82576 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.
Reserved	23	0b	Reserved
RTLC	24	0b	Re-transmit on Late Collision When set, enables the 82576 to re-transmit on a late collision event.
Reserved	25	0b	Reserved
Reserved	27:26	0x1	Reserved
Reserved	31:28	0xA	Reserved

8.12.2 Transmit Control Extended - TCTL_EXT (0x0404; R/W)

This register controls late collision detection.

COLD is used to determine the latest time in which a collision indication is considered as a valid collision and not a late collision. When using the internal PHY, the default value of 0x40 provides a behavior consistent with the 802.3 spec requested behavior. However, when using an SGMII connected PHY, the SGMII adds some delay on top of the time budget allowed by the specification (collisions in valid network topographies even after 512 bit time can be expected). In order to accommodate this condition, COLD should be updated to take the SGMII inbound and outbound delays. The delay induced by the 82576 is 16 bit time in 10 Mb/s (add 2 to the COLD field value) and 40 bit time in 100 Mb/s (add 5 to the COLD field value). Any delay induced by the specific PHY used should also be added.

Field	Bit(s)	Initial Value	Description
Reserved	9:0	0x40	Reserved
COLD	19:10	0x42	Collision Distance (in byte time) Used to determine the latest time in which a collision indication is considered as a valid collision and not a late collision.
Reserved	31:20	0x0	Reserved.

8.12.3 Transmit IPG Register - TIPG (0x0410; R/W)

This register controls the Inter Packet Gap (IPG) timer.

Field	Bit(s)	Initial Value	Description
IPGT	9:0	0x08	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. IPGT specifies the IPG length for back-to-back transmissions in both full duplex and half duplex. Note that an offset of 4 byte times is added to the programmed value to determine the total IPG. As a result, a value of 8 is recommended to achieve a 12 byte time IPG.
IPGR1	19:10	0x04	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.
IPGR	29:20	0x06	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	00b	Reserved Read as 0b. Should be written with 0b for future compatibility.

8.12.4 DMA Tx Control - DTXCTL (0x03590; R/W)

This register is used to set some parameters controlling the DMA Tx behavior.

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved.
NoSnoop_LSO_hdr_buf	1	0b	NoSnoop Header buffer of TSO packets. In TSO packets, the header is fetched again for each segment sent. When this bit is set, the header buffer fetching for all segments, apart from the first one, sets no-snoop mode. When reset, all the header buffer fetching uses the attribute as in the DCA registers. The first segment always uses the attribute as in the DCA registers.
8023LL	2	1b	802.3 length location 1b = The location of the 802.3 length field in 802.3+SNAP packets, is assumed to be 8 bytes before the end of the MAC header. 0b = The location of the 802.3 length field in 802.3+SNAP packets, is calculated from the beginning of the MAC header assuming no VLAN present in the packet sent by the software. This bit is used only in case of large send (TSO) with SNAP mode.
Add VLAN location	3	0b	1b = Added by MAC - means that Loopbacked packets are without VLAN. 0b = Added by DMA - means that Loopbacked packets are with VLAN.
OutOfSyncEnable	4	0b	0b = Out Of sync mechanism is disabled. 1b = Out Of sync mechanism is enabled.
MDP_EN	5	0b	Malicious driver protection enable 0b = mechanism is disabled. 1b = mechanism is enabled.
SPOOF_INT	6	1	Interrupt on spoof behavior detection 0b = mechanism is disabled. 1b = mechanism is enabled.
Reserved	31:7	0x0	Reserved

8.12.5 DMA TX TCP Flags Control Low - DTXTCPFLGL (0x359C; RW)

This register holds the buses that “AND” the control flags in TCP header for the first and middle segments of a TSO packet. See Section 7.2.4.7.1 and Section 7.2.4.7.2 for details on the use of this register.

Field	Bit(s)	Initial Value	Description
TCP_flg_first_seg	11:0	0xFF6	TCP Flags first segment. Bits that make AND operation with the TCP flags at TCP header in the first segment
Reserved	15:12	0x00	Reserved
TCP_Flg_mid_seg	27:16	0x76	TCP Flags middle segments. The low bits that make AND operation with the TCP flags at TCP header in the middle segments
Reserved	31:28	0x00	Reserved

8.12.6 DMA TX TCP Flags Control High - DTXTCPFLGH (0x35A0; RW)

This register holds the buses that “AND” the control flags in TCP header for the last segment of a TSO packet. See Section 7.2.4.7.3 for details of use of this register

Field	Bit(s)	Initial Value	Description
TCP_Flg_lst_seg	11:0	0xF7F	TCP Flags last segment. Bits that make AND operation with the TCP flags at TCP header in the last segment
Reserved	31:12	0x00	Reserved.

8.12.7 DMA TX Max Total Allow Size Requests - DTXMXSZRQ (0x3540; RW)

This register limits the total number of data bytes that might be in outstanding PCIe requests from the host memory. This allows requests to send low latency packets to be serviced in a timely manner, as this request is serviced right after the current outstanding requests are completed.

Field	Bit(s)	Initial Value	Description
Max_bytes_num_req	11:0	0x10	Max allowed number of bytes requests. The maximum allowed amount of 256 bytes outstanding requests. If the total size request is higher than the amount in the field no arbitration is done and no new packet is requested.
Reserved	31:12	0x0	Reserved

8.12.8 Transmit Descriptor Base Address Low - TDBAL (0xE000 + 0x40*n [n=0...15]; R/W)

These registers contain the lower 32 bits of the 64-bit descriptor base address. The lower 7 bits are ignored. The Transmit Descriptor Base Address must point to a 128-byte aligned block of data.

Field	Bit(s)	Initial Value	Description
0	6:0	0x0	Ignored on writes. Returns 0b on reads.
TDBAL	31:7	X	Transmit Descriptor Base Address Low

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3800, 0x3900, 0x3A00 & 0x3B00 respectively.

8.12.9 Transmit Descriptor Base Address High - TDBAH (0x0E004 + 0x40*n [n=0...15]; R/W)

These registers contain the upper 32 bits of the 64-bit descriptor base address.

Field	Bit(s)	Initial Value	Description
TDBAH	31:0	X	Transmit Descriptor Base Address [63:32]

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3804, 0x3904, 0x3A04 & 0x3B04 respectively.

8.12.10 Transmit Descriptor Ring Length - TDLEN (0x0E008 + 0x40*n [n=0...15]; R/W)

These registers contain the descriptor ring length. The registers indicates the length in bytes and must be 128-byte aligned.

Field	Bit(s)	Initial Value	Description
Reserved	6:0	0x0	Must be set to zero.
LEN	19:7	0x0	Descriptor Ring Length (number of 8 descriptor sets)
Reserved	31:20	0x0	Reserved Reads as 0b. Should be written to 0b.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3808, 0x3908, 0x3A08 & 0x3B08 respectively.

8.12.11 Transmit Descriptor Head - TDH (0x0E010 + 0x40*n [n=0...15]; R/W)

These registers contain the head pointer for the transmit descriptor ring. It points to a 16-byte datum. Hardware controls this pointer.

Note: The values in these registers might point to descriptors that are still not in host memory. As a result, the host cannot rely on these values in order to determine which descriptor to release.

Field	Bit(s)	Initial Value	Description
TDH	15:0	0x0	Transmit Descriptor Head
Reserved	31:16	0x0	Reserved Should be written to 0b.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3810, 0x3910, 0x3A10 & 0x3B10 respectively.

8.12.12 Transmit Descriptor Tail - TDT (0x0E018 + 0x40*n [n=0...15]; R/W)

These registers contain the tail pointer for the transmit descriptor ring and 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.

Field	Bit(s)	Initial Value	Description
TDT	15:0	0x0	Transmit Descriptor Tail
Reserved	31:16	0x0	Reserved Reads as 0b. Should be written to 0b for future compatibility.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3818, 0x3918, 0x3A18 & 0x3B18 respectively.

8.12.13 Transmit Descriptor Control - TXDCTL (0x0E028 + 0x40*n [n=0...15]; R/W)

These registers control the fetching and write-back of transmit descriptors. The three threshold values are used to determine when descriptors are read from and written to host memory. The values are in units of descriptors (each descriptor is 16 bytes).

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 when *RS* is set. In addition, with transmit descriptor bursting enabled, some descriptors are written back that did not have *RS* set in their respective descriptors.

Note: When WTHRESH = 0b, only descriptors with the *RS* bit set are written back

Field	Bit(s)	Initial Value	Description
PTHRESH	4:0	0x0	Prefetch Threshold Controls when a prefetch of descriptors is considered. This threshold refers to the number of valid, unprocessed transmit descriptors the 82576 has in its on-chip buffer. If this number drops below PTHRESH, the algorithm considers pre-fetching descriptors from host memory. However, this fetch does not happen unless there are at least HTHRESH valid descriptors in host memory to fetch. **Note:** HTHRESH should be given a non zero value each time PTHRESH is used.
Reserved	7:5	0x0	Reserved
HTHRESH	12:8	0x0	Host Threshold
Reserved	15:13	0x0	Reserved Reads as 0b. Should be written as 0b for future compatibility.
WTHRESH	20:16	0x0	Write-Back Threshold Controls the write-back of processed transmit descriptors. This threshold refers to the number of transmit descriptors in the on-chip buffer that are ready to be written back to host memory. In the absence of external events (explicit flushes), the write-back occurs only after at least WTHRESH descriptors are available for write-back. **Note:** Since the default value for write-back threshold is 0b, descriptors are normally written back as soon as they are processed. WTHRESH must be written to a non-zero value to take advantage of the write-back bursting capabilities of the 82576.
Reserved	24:21	0x0	Reserved
ENABLE	25	1b/0b	Transmit Queue Enable When set, this bit enables the operation of a specific transmit queue: Default value for Q0 = 1b. Default value for Q15:1 = 0b. After a VF FLR to VF0, Q0 is also reset to zero. Setting this bit initializes all the internal registers of a specific queue. Until then, the state of the queue is kept and can be used for debug purposes. When disabling a queue, this bit is cleared only after all activity at the queue stopped. Note: This bit is valid only if the queue is actually enabled, thus if TCTL.TXEN is cleared, this bit remains zero.
SWFLSH	26	0b	Transmit Software Flush This bit enables software to trigger descriptor write-back flushing, independently of other conditions. This bit is self cleared by hardware.
Reserved	27	0b	Reserved (was Priority)

Field	Bit(s)	Initial Value	Description
Reserved	28	0B	Reserved
Reserved	29	0	Reserved
Reserved	31:30	0x00	Reserved

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3828, 0x3928, 0x3A28 and 0x3B28 respectively.

8.12.14 Tx Descriptor Completion Write–Back Address Low - TDWBAL (0x0E038 + 0x40*n [n=0...15]; R/W)

Field	Bit(s)	Initial Value	Description
Head_WB_En	0	0b	Head Write-Back Enable 1b = Head write-back is enabled. 0b = Head write-back is disabled. When head_WB_en is set, SN_WB_en is ignored and no descriptor write-back is executed.
WB on EITR	1	0b	When set, a head write back is done upon EITR expiration.
HeadWB_Low	31:2	0x0	Lowest 32 bits of the head write-back memory location (DWORD aligned). Last 2 bits of this field are ignored and are always read as 0.0, meaning that the actual address is QWORD aligned.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3838, 0x3938, 0x3A38 & 0x3B38 respectively.

8.12.15 Tx Descriptor Completion Write–Back Address High - TDWBAH (0x0E03C + 0x40*n [n=0...15];R/W)

Field	Bit(s)	Initial Value	Description
HeadWB_High	31:0	0x0	Highest 32 bits of the head write-back memory location.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x383C, 0x393C, 0x3A3C & 0x3B3C respectively.

8.13 DCA Register Descriptions

8.13.1 Rx DCA Control Registers - RXCTL (0x0C014 + 0x40*n [n=0...15]; R/W)

Note: RX data write no-snoop is activated when the NSE bit is set in the receive descriptor.

Field	Bit(s)	Initial Value	Description
Reserved	4:0	0x0	Reserved
RX Descriptor DCA EN	5	0b	Descriptor DCA Enable When set, hardware enables DCA for all Rx descriptors written back into memory. When cleared, hardware does not enable DCA for descriptor write-backs. This bit is cleared as a default.
Rx Header DCA EN	6	0b	Rx Header DCA Enable When set, hardware enables DCA for all received header buffers. When cleared, hardware does not enable DCA for Rx headers. This bit is cleared as a default.
Rx Payload DCA EN	7	0b	Payload DCA Enable When set, hardware enables DCA for all Ethernet payloads written into memory. When cleared, hardware does not enable DCA for Ethernet payloads. This bit is cleared as a default.
RXdescRead NSEn	8	0b	Rx Descriptor Read No Snoop Enable This bit must be reset to 0b to ensure correct functionality (Except if the software driver can guarantee the data is present in the main memory before the DMA process occur).
RXdescRead ROEn	9	1b	Rx Descriptor Read Relax Order Enable
RXdescWBNSen	10	0b	Rx Descriptor Write-Back No Snoop Enable This bit must be reset to 0b to ensure correct functionality of descriptor write-back.
RXdescWBROen (RO)	11	0b	Rx Descriptor Write-Back Relax Order Enable This bit must be reset to 0b to ensure correct functionality of descriptor write-back.
RXdataWrite NSEn	12	0b	Rx Data Write No Snoop Enable (header replication: header and data) When set to 0b, the last bit of the *Packet Buffer Address* field in the advanced receive descriptor is used as the LSB of the packet buffer address (A0), thus enabling 8-bit alignment of the buffer. When set to 1b, the last bit of the *Packet Buffer Address* field in advanced receive descriptor is used as the No-Snoop Enabling (NSE) bit (buffer is 16-bit aligned). If also set to 1b, the NSE bit determines whether the data buffer is snooped or not.
RXdataWrite ROEn	13	1b	Rx Data Write Relax Order Enable (header replication: header and data)
RxRepHeader NSEn	14	0b	Rx Replicated/Split Header No Snoop Enable This bit must be reset to 0b to ensure correct functionality of header write to host memory.
RxRepHeader ROEn	15	1b	Rx Replicated/Split Header Relax Order Enable
Reserved	23:16	0b	Reserved
CPUID	31:24	0x0	PPhysical ID Legacy DCA capable platforms - the device driver, upon discovery of the physical CPU ID and CPU Bus ID, programs the CPUID field with the Physical CPU and Bus ID associated with this Rx queue. DCA 1.0 capable platforms - the device driver programs a value, based on the relevant APIC ID, associated with this Tx queue. See Table 3.1.3.1.2.3 for details

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2814, 0x2914, 0x2A14 & 0x2B14 respectively.

8.13.2 Tx DCA Control Registers - TXCTL (0x0E014 + 0x40*n [n=0...15]; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	4:0	0	Reserved
TX Descriptor DCA EN	5	0b	Descriptor DCA Enable When set, hardware enables DCA for all Tx descriptors written back into memory. When cleared, hardware does not enable DCA for descriptor write-backs. This bit is cleared as a default and also applies to head write-back when enabled.
Reserved	7:6	00b	Reserved
TXdescRDNSen	8	0b	Tx Descriptor Read No Snoop Enable This bit must be reset to 0b to ensure correct functionality (unless the software device driver has written this bit with a write-through instruction).
TXdescRDROEn	9	1b	Tx Descriptor Read Relax Order Enable
TXdescWBNSen	10	0b	Tx Descriptor Write-Back No Snoop Enable This bit must be reset to 0b to ensure correct functionality of descriptor write-back. Also applies to head write-back, when enabled.
TXdescWBROen	11	1b	Tx Descriptor Write-Back Relax Order Enable Applies to head write-back, when enabled.
TXDataReadNSEn	12	0b	Tx Data Read No Snoop Enable
TXDataReadROEn	13	1b	Tx Data Read Relax Order Enable
Reserved	23:14	0	Reserved
CPUID	31:24	0x0	Physical ID Legacy DCA capable platforms - the device driver, upon discovery of the physical CPU ID and CPU Bus ID, programs the CPUID field with the Physical CPU and Bus ID associated with this Tx queue. DCA 1.0 capable platforms - the device driver programs a value, based on the relevant APIC ID, associated with this Tx queue. See Table 3.1.3.1.2.3 for details

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3814, 0x3914, 0x3A14 & 0x3B14 respectively.

8.13.3 DCA Requester ID Information - DCA_ID (0x05B70; RO)

The DCA requester ID field, composed of Device ID, Bus #, and Function # is set up in MMIO space for software to program the DCA Requester ID Authentication register.

Field	Bit(s)	Initial Value	Description
Function Number	2:0	000b	Function Number Function number assigned to the function based on BIOS/operating system enumeration.

Device Number	7:3	0x0	Device Number Device number assigned to the function based on BIOS/ operating system enumeration.
Bus Number	15:8	0x0	Bus Number Bus number assigned to the function based on BIOS/operating system enumeration.
Reserved	31:16	0x0	Reserved

8.13.4 DCA Control - DCA_CTRL (0x05B74; R/W)

Field	Bit(s)	Initial Value	Description
DCA_DIS	0	1b	DCA Disable 0b = DCA tagging is enabled for this port. 1b = DCA tagging is disabled for this port.
DCA_MODE	4:1	0x0	DCA Mode 000b = Legacy DCA is supported. The TAG field in the TLP header is based on the following coding: bit 0 is DCA enable; bits 3:1 are CPU ID). 001b = DCA 1.0 is supported. When DCA is disabled for a given message, the TAG field is 0000,0000b. If DCA is enabled, the TAG is set per queue as programmed in the relevant DCA Control register. All other values are undefined.
Reserved	31:5	0x0	Reserved

8.14 Virtualization Register Descriptions

For all the registers in this section, VT_CTL replaces the VMD_CTL register of the 82575.

8.14.1 Next Generation VMDq Control register – VT_CTL (0x0581C; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	6:0	0x0	Reserved
DEF_PL	9:7	00b	Default pool - used to queue packets that did not pass any VM queuing decision.
Reserved	26:10	0x0	Reserved
FLP	27	0b	Filter local packets - filter incoming packets whose MAC source address matches one of the incoming DA MAC addresses. If the SA of the received packet matches one of the DA in the RAH/RAL registers, then the VM tied to this DA does not receive the packet. Other VMs can still receive it.
IGMAC	28	0x0	If set, MAC address is ignored during pool decision. Pooling is based on VLAN only. If this bit is set, then the VMOLR.strvlan should be set to the same value for all pools,
Dis_Def_Pool	29	0x0	Drop if no poll is found. If this bit is asserted, then in a RX switching, in a virtualized environment, if there is no destination pool, the packet is discarded and not sent to the default pool. Otherwise, it is sent to the pool defined by the DEF_PL field.
Rpl_En	30	0x0	Replication Enable
Reserved	31	0x0	Reserved

8.14.2 Physical Function Mailbox - PFMailbox (0x0C00 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Sts (WO)	0	0b	Status/Command from PF ready. Setting this bit, causes an interrupt to the relevant VF. This bit always read as zero. Setting this bit sets the PFSTS bit in VFMailbox.
Ack (WO)	1	0b	VF message received. Setting this bit, causes an interrupt to the relevant VF. This bit always read as zero. Setting this bit sets the PFACK bit in VFMailbox.
VFU	2	0b	Buffer is taken by VF. This bit is RO for the PF and is a mirror of the VFU bit of the VFMailbox register.
PFU	3	0b	Buffer is taken by PF. This bit can be set only if the VFU bit is cleared and is mirrored in the PFU bit of the VFMailbox register.
RVFU (WO)	4	0b	Reset VFU - setting this bit clears the VFU bit in the corresponding VFMailbox register - this bit should be used only if the VF driver is stuck. Setting this bit is also reset the corresponding bits in the MBVFICR VFREQ & VFACK fields.
Reserved	31:5	0x0	Reserved

The usage of the mailbox register set is described in Section 7.9.2.9.1.

8.14.3 Virtual Function Mailbox - VFMailbox (0x0C40 + 4*n [n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Req (WO)	0	0b	Request for PF ready. Setting this bit, causes an interrupt to the PF. This bit always read as zero. Setting this bit sets the corresponding bit in VFREQ field in MBVFICR register.
Ack (WO)	1	0b	PF message received. Setting this bit, causes an interrupt to the PF. This bit always read as zero. Setting this bit sets the corresponding bit in VFACK field in MBVFICR register.
VFU	2	0b	Buffer is taken by VF. This bit can be set only if the PFU bit is cleared and is mirrored in the VFU bit of the PFMailbox register.
PFU	3	0b	Buffer is taken by PF. This bit is RO for the VF and is a mirror of the PFU bit of the PFMailbox register.
PFSTS (RC)	4	0b	PF wrote a message in the mailbox.
PFACK (RC)	5	0b	PF acknowledged the VF previous message.
RSTI	6	1b	Indicates that the PF had reset the shared resources and the reset sequence is in progress.
RSTD (RC)	7	0b	Indicates that a PF software reset completed and the VF can start to use the device.
Reserved	31:8	0x0	Reserved

8.14.4 Virtualization Mailbox Memory - VMBMEM (0x0800:0x083C + 0x40*n [n=0...7]; R/W)

Mailbox memory for PF and VF drivers communication. Locations can be accessed as 32-bit or 64-bit words.

The memory is accessible to the PF and the VFs according to the following mapping.

RAM address	Function	PF BAR 0 mapping[1]	VF BAR 0 mapping
0 - 63	VF0 ↔ PF	0 - 63	VMBMEM:VMBMEM + 63
64 - 127	VF1 ↔ PF	64 - 127	VMBMEM:VMBMEM + 63
….			
384 - 447	VF7↔ PF	384 - 447	VMBMEM:VMBMEM + 63

[1] Relative to VMBMEM register.

Field	Bit(s)	Initial Value	Description
Mailbox Data	31:0	X	Mailbox Data

8.14.5 Mailbox VF Interrupt Causes Register - MBVFICR (0x0C80; R/W1C)

Field	Bit(s)	Initial Value	Description
VFREQ	7:0	0x0	VF #n wrote a message.

Field	Bit(s)	Initial Value	Description
Reserved	15:8	0x0	Reserved
VFACK	23:16	0x0	VF #n acknowledged a PF message.
Reserved	31:24	0x0	Reserved

8.14.6 Mailbox VF Interrupt Mask Register - MBVFIMR (0x0C84; RW)

Field	Bit(s)	Initial Value	Description
VFIM	7:0	0xFF	Mailbox indication from VF #n can cause an interrupt to the PF..
Reserved	31:8	0x0	Reserved

8.14.7 FLR Events - VFLRE (0x0C88; R/W1C)

This register reflects the VFLR events of the different VFs. It is accessible only to the PF. These bits are cleared by writing 1.

Field	Bit(s)	Initial Value	Description
VFLR	7:0	X	Reflects a VFLR event in VF7 to VF0 respectively.
Reserved	31:8	0x0	Reserved

8.14.8 VF Receive Enable- VFRE (0x0C8C; RW)

Field	Bit(s)	Initial Value	Description
VFRE	7:0	0xFF	Enables filtering process to forward packets to VF7 to VF0 respectively. Each bit is cleared by the relevant VFLR or by a VF SW reset.
Reserved	31:8	0x0	Reserved

8.14.9 VF Transmit Enable - VFTE (0x0C90; RW)

Field	Bit(s)	Initial Value	Description
VFTE	7:0	0xFF	Enables transmit process to forward packets from VF7 to VF0 respectively. Each bit is cleared by the relevant VFLR or by a VF SW reset.
Reserved	31:8	0x0	Reserved

Note: Clearing one of VFTE bits may cause a transmit packet drop from the disabled queue.

8.14.10 Wrong VM Behavior Register - WVBR (0x3554; RC/W1C)

Field	Bit(s)	Initial Value	Description
WVM	15:0	0x0	Bitmap indicating against which queue an anti spoof action was taken
Reserved	31:16	0x0	Reserved

8.14.11 VM Error Count Mask – VMECM (0x3510; RW)

Field	Bit(s)	Initial Value	Description
Filter	7:0	0x0	Defines if a packet dropped from pools 0 to 7 respectively is counted in the SSVPC counter.
Reserved	31:8	0x0	Reserved

8.14.12 Last VM Misbehavior Cause – LVMMC (0x3548; RC)

Field	Bit(s)	Initial Value	Description
MAC spoof	0	0b	A MAC spoof attempt was detected
VLAN spoof	1	0b	A VLAN spoof attempt was detected
Legacy desc in RT/IOV	2	0b	A legacy desc in RT/IOV was detected
Out of sync - single send	3	0b	An out of sync misbehavior was detected in a single send operation.
Out of sync - large send	4	0b	An out of sync misbehavior was detected in a large send operation.
Reserved	11:5	0b	Reserved
L3 Type	12	0b	0 = The error was detected in an IPv6 packet 1 = The error was detected in an IPv4 packet
L4 Type	14:13	0b	Indicates the L4 type of the erroneous packet: 00b: UDP 01b: TCP 10b: SCTP 11b: Reserved
Reserved	15	0b	Reserved
Queue	19:16	0x0	Queue in which the illegal behavior was detected
Reserved	31:20	0x0	Reserved

8.14.13 Queue drop Enable Register - QDE (0x2408;RW)

This register allows the PF to override the SRRCTL.drop_en bit set by the VF, to avoid head of line blocking issues if an un-trusted VF does not provide a receive descriptor to the hardware.

Field	Bit(s)	Initial Value	Description
QDE	15:0	0x0	Enable drop packets from queue 15 to 0 respectively. This bit overrides the SRRCTL.drop_en bit of each queue. If either of the bits is set, a packet received when no is descriptor available is dropped.
Reserved	31:16	0x0	Reserved

8.14.14 DMA Tx Switch control - DTXSWC (0x3500; R/W)

This register controls the security settings of the switch and enables the loopback mode.

Field	Bit(s)	Initial Value	Description
MACAS	7:0	0x0	Enable anti spoofing filter on MAC addresses for VF7 to VF0 respectively.
VLANAS	15:8	0x0	Enable anti spoofing filter on VLAN tags for VF7 to VF0 respectively.
LLE	23:16	0x0	Local loopback enable - when set, a packet originating from pool n and destined to pool n is looped back. If clear, the packet is dropped.
Reserved	30:24	0x0	Reserved
Loopback_en	31	0b	Enable Next Generation VMDq loopback.

8.14.15 VM VLAN Insert Register – VMVIR (0x3700 + 4 *n [n=0..7]; RW)

Field	Bit(s)	Initial Value	Description
Port VLAN ID	15:0	0x0	Port VLAN tag to insert in case action = 1.
Reserved	29:16	0x0	Reserved
vlana	31:30	0x0	VLAN action: 00b = use descriptor command 01b = always insert Default VLAN 10b = Never insert VLAN 11b = Reserved.

8.14.16 VM Offload Register - VMOLR (0x05AD0 + 4*n [n=0...7]; RW)

This register controls the offload and queueing options applied to each VF.

Field	Bit(s)	Initial Value	Description
rlpml	13:0	0x2600	Long packet size (9k default)
Reserved	15:14	0x0	Reserved
lpe	16	0b	Long packet enable
RSSE	17	0b	0b = When in RSS + Next Generation VMDq mode (MRQC.Multiple Receive Queues Enable = 101b) packets for this VM is forwarded to the pool's default queue as defined in VT_CTL.Default Next Generation VMDq Queue. 1b = When in RSS + mode (MRQC.Multiple Receive Queues Enable = 101b) packets for this VM is forwarded according to RSS redirection table (RETA).
Reserved	23:18	0x0	Reserved

aupe	24	0b	Accept Untagged packets enable. When set, packets without VLAN tag can be forwarded to this queue, assuming they pass the MAC address queueing mechanism.
rompe	25	0x0	receive overflow multicast packets - accept packets that match the MTA table.
rope	26	0x0	receive overflow packets - accept packets that match the UTA table.
bam	27	0x0	Broadcast accept
mpe	28	0x0	multicast promiscuous
Reserved	29	0x0	Reserved
strvlan	30	0x0	VLAN strip
Reserved	31	0x1	Reserved - must be set to one. .

8.14.17 Replication Offload Register - RPLOLR (0x05AF0; RW)

This register describes the off loads applied to multicast packets.

Field	Bit(s)	Initial Value	Description
Reserved	29:0	0x0	Reserved
strvlan	30	0x0	VLAN strip
strcrc	31	0x1	Reserved

8.14.18 VLAN VM Filter - VLVF (0x05D00 + 4*n [n=0...31]; RW)

This register set describes which VLANs the local VMs are part of. Each register contains a VLAN tag and a list of the VFs which are part of it. Only packets with a VLAN matching one of the VLAN tags of which the VF is member of are forwarded to this VF.

Field	Bit(s)	Initial Value	Description
VLAN_Id	11:0	0x0	Defines a VLAN tag to which each VM whose bit is set in the POOLSEL field is set belongs.
POOLSEL	19:12	0x0	Pool Select (bitmap)
LVLAN	20	0x0	This VLAN is local and packets with this VLAN should not be forwarded to the external NIC.
Reserved	30:21	0x0	
VI_En	31	0b	VLAN Id Enable - this filter is valid.

8.14.19 Unicast Table Array - UTA (0xA000 + 4*n [n=0...127]; WO)

There is one register per 32 bits of the Unicast Address Table for a total of 128 registers (the UTA[127:0] designation). Software must mask to the desired bit on reads and supply a 32-bit word on writes. The first bit of the address used to access the table is set according to the RCTL.MO field.

Note: All accesses to this table must be 32 bit.

The lookup algorithm is the same one used for the MTA table.

This table should be zeroed by software before start of work.

The data returned when reading this table is unexpected.

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Word wide bit vector specifying 32 bits in the unicast destination address filter table.

8.14.20 Storm Control Control Register- SCCRL (0x5DB0;RW)

Field	Bit(s)	Initial Value	Description
MDIPW	0	0b	Drop multicast packets (excluding flow control and manageability packets) if multicast threshold is exceeded in previous window
MDICW	1	0b	Drop multicast packets (excluding flow control and manageability packets) if multicast threshold is exceeded in current window
BDIPW	2	0b	Drop broadcast packets (excluding flow control and manageability packets) if broadcast threshold is exceeded in previous window
BDICW	3	0b	Drop broadcast packets (excluding flow control and manageability packets) if broadcast threshold is exceeded in current window
BIDU	4	0b	BSC Include Destination Unresolved: If bit is set, unicast received packets with no destination pool and sent to the default pool is included in IBSC
RSVD	7:5	0x0	Reserved
INTERVAL	17:8	0x8	BSC/MSC Time-interval-specification: The interval size for applying Ingress Broadcast or Multicast Storm Control. Interrupt decisions are made at the end of each interval (and most flags are also set at interval end). Setting this field resets the counter.
RSVD	31:18	0x0	Reserved

8.14.21 Storm Control Status - SCSTS (0x5DB4;RO)

Field	Bit(s)	Initial Value	Description
BSCA	0	0b	Broadcast storm control active
BSCAP	1	0b	Broadcast storm control active in previous window
MSCA	2	0b	Multicast storm control active
MSCAP	3	0b	Multicast storm control active in previous window
RSVD	31:4	0x0	Reserved

8.14.22 Broadcast Storm Control Threshold - BSCTRH (0x5DB8;RW)

Field	Bit(s)	Initial Value	Description
UTRESH	18:0	0x0	Traffic Upper Threshold-size: Represents the upper threshold for broadcast storm control.
RSVD	31:19	0x0	Reserved

8.14.23 Multicast Storm Control Threshold - MSCTRH (0x5DBC; RW)

Field	Bit(s)	Initial Value	Description
UTRESH	18:0	0x0	Traffic Upper Threshold-size: Represents the upper threshold for multicast storm control.
RSVD	31:19	0x0	Reserved

8.14.24 Broadcast Storm Control Current Count - BSCCNT (0x5DC0;RO)

Field	Bit(s)	Initial Value	Description
CCOUNT	24:0	0x0	IBSC Traffic Current Count: Represents the count of broadcast traffic received in the current time interval in units of 64-byte segments.
RSVD	31:25	0x0	Reserved.

8.14.25 Multicast Storm Control Current Count - MSCCNT (0x5DC4;RO)

Field	Bit(s)	Initial Value	Description
CCOUNT	24:0	0x0	IMSC Traffic Current Count: Represents the count of multicast traffic received in the current time interval in units of f 64-byte segments.
RSVD	31:25	0x0	Reserved.

8.14.26 Storm Control Time Counter - SCTC (0x5DC8; RO)

This register keeps track of the number of time units elapsed since the end of last time interval.

Field	Bit(s)	Initial Value	Description
COUNT	9:0	0x0	SC Time Counter: The counter for number of time units elapsed since the end of the last time interval.
RSVD	31:10	0x0	Reserved.

8.14.27 Storm Control Basic Interval- SCBI (0x5DCC; RW)

This register defines the basic interval used as the base for the SCCRL.Interval counting in 10 Mb/s speed. This register is defined in 16 ns clock cycles. The interval in 1000/100 is 100 or 10 time smaller respectively.

Field	Bit(s)	Initial Value	Description
BI	24:0	0x5F5E10	Basic interval
RSVD	31:25	0x0	Reserved.

8.14.28 Virtual Mirror Rule Control - VMRCTL (0x5D80 + 0x4*n [n= 0..3]; RW)

This register controls the rules to be applied and the destination port.

Field	Bit(s)	Initial Value	Description
VPME	0	0b	Virtual pool mirroring enable- reflects all the packets sent to a set of given VMs.
UPME	1	0b	Uplink port mirroring enable - reflects all the traffic received from the network.
DPME	2	0b	Downlink port mirroring enable - reflects all the traffic transmitted to the network.
VLME	3	0b	VLAN mirroring enable - reflects all the traffic received in a set of given VLANs. Either from the network or from local VMs.
Reserved	7:4	0x0	Reserved
MP	10:8	0x0	VM Mirror port destination.
Reserved	31:11	0x0	Reserved

8.14.29 Virtual Mirror Rule VLAN - VMRVLAN (0x5D90 + 0x4*n [n= 0..3]; RW)

This register controls the VLAN ports as listed in the VLVF table taking part in the VLAN mirror rule.

Field	Bit(s)	Initial Value	Description
VLAN	31:0	0x0	Bitmap listing which VLANs participate in the mirror rule

8.14.30 Virtual Mirror Rule VM - VMRVM (0x5DA0 + 0x4*n [n= 0..3]; RW)

This register controls the VLAN ports as listed in the VLVF table taking part in the VLAN mirror rule.

Field	Bit(s)	Initial Value	Description
VM	7:0	0x0	Bitmap listing which VMs participate in the mirror rule
Reserved	31:8	0x0	Reserved

8.15 Tx Bandwidth Allocation to VM Register Description

These registers are owned by the PF in an IOV mode.

8.15.1 VM Bandwidth Allocation Control & Status - VMBACS (0x3600; RW)

Field	Bit(s)	Initial Value	Description
8BYTE_VAL	7:0	0x04	8-Bytes Time Counter Value in DMA clocks unit. Link speed of 1Gbps - must be set to 0x04 Link speed of 100Mbps - must be set to 0x28
Reserved	15:8	0x08	Reserved
VMBA_EN1	19:16	0	VM Bandwidth Allocation Enable field 1. 0x0 – for non-virtualized contexts. 0x7 – for virtualized contexts.
VMBA_SET	20	0, RO	VM Bandwidth Allocation is set. (RO) When set, it indicates that at least one queue is currently rate-controlled for achieving the bandwidth allocation scheme to VMs. Used by SW for the link speed change procedure. When cleared, the VM rate-controllers are all disabled.
Reserved	23:21	010b	Reserved - must be set to its initial value.
VMBA_EN2	27:24	0x0	VM Bandwidth Allocation Enable field 2. 0x0 – for non-virtualized contexts. 0xF – for virtualized contexts.
Reserved	30:28	0	Reserved
SPEED_CHG	31	0 / Read & Clear only	Link Speed has changed. Set by HW to indicate that the link speed has changed. Cleared by SW at the end of the link speed change procedure.

8.15.2 VM Bandwidth Allocation Max Memory Window - VMBAMMW (0x3670; RW)

Field	Bit(s)	Initial Value	Description
MMW_SIZE	10:0	0	Max Memory Window Size for the VM rate-controllers. It is the maximum amount of 1KB units of transmit compensation payload that can be accumulated for a Tx queue.
Reserved	31:11	0	Reserved

8.15.3 VM Bandwidth Allocation Select - VMBASEL (0x3604; RW)

Field	Bit(s)	Initial Value	Description
TXQ_IDX	3:0	0	Tx Queue Index. This register is used to select the Tx Queue for which the bandwidth share configuration will be set. Prior to write access the VMRSC register, software has to set this field with the index of the Tx queue to be accessed.
Reserved	31:4	0	Reserved

8.15.4 VM Bandwidth Allocation Config - VMBAC (0x3608; RW)

Field	Bit(s)	Initial Value	Description
RF_DEC	13:0	X	VM Rate Factor Hexadecimal Part, for the Tx queue indexed by TXQ_IDX field in VMBASEL register. Rate factor bits that come after the hexadecimal point. Rate Factor (RF) is defined as the ratio between the link speed (1 Gb/s or 100Mbps) and the rate allocated to that VM. Assign RF to VMs so that: SUM(VM rates) = link speed, i.e. 1 Gb/s or 100Mbps. Minimum allowed bandwidth share for a VM is 0.1% of the link speed. Limit the maximum Rate Factor accordingly. Meaningful only if RC_ENA bit is set.
RF_INT	23:14	X	VM Rate Factor Integer Part, for the Tx queue indexed by TXQ_IDX field in VMBASEL register. Rate factor bits that come before the hexadecimal point. Meaningful only if RC_ENA bit is set.
Reserved	30:24	0	Reserved
RC_ENA	31	0 RW / RO if VT is fused-off	VM Rate-Controller Enable, for the Tx queue indexed by TXQ_IDX field in VMBASEL register. When set, the bandwidth share allocated to the VM by programming this register is enforced, used for virtualized contexts. When cleared, other fields in this register are meaningless. The VM operates in the “Bandwidth Takeover” mode, taking over the link’s bandwidth left unused by others. For non-virtualized contexts, this bit must be cleared for all the Tx Queues.

8.16 Timer Register Descriptions

8.16.1 Watchdog Setup - WDSTP (0x01040; R/W)

Field	Bit(s)	Initial Value	Description
WD_Enable	0	0b[1]	Enable Watchdog Timer
WD_Timer_ Load_enable (SC)	1	0b	Enables the load of the watchdog timer by writing to WD_Timer field. If this bit is not set, the WD_Timer field is loaded by the value of WD_Timeout. **Note:** Writing to this field is only for DFX purposes.
Reserved	15:2	0x0	Reserved
WD_Timer (RWS)	23:16	WD_Timeout	Indicates the current value of the timer. Resets to the timeout value each time the 82576 functional bit in Software Device Status register is set. If this timer expires, the WD interrupt to the firmware and the WD SDP is asserted. As a result, this timer is stuck at zero until it is re-armed. **Note:** Writing to this field is only for DFX purposes.
WD_Timeout	31:24	0x0[1]	Defines the number of seconds until the watchdog expires. The granularity of this timer is 1 sec. The minimal value allowed for this register when the watchdog mechanism is enabled is two. Setting this field to 1b might cause the watchdog to expire immediately.

[1] Value read from the EEPROM.

8.16.2 Watchdog Software Device Status - WDSWSTS (0x01044; R/W)

Field	Bit(s)	Initial Value	Description
Dev_Functional (SC)	0	0b	Each time this bit is set, the watchdog timer is re-armed. This bit is self clearing
Force_WD (SC)	1	0b	Setting this bit causes the WD timer to expire immediately. The WD_timer field is set to 0b. It can be used by software in order to indicate some fatal error detected in the software or in the hardware. This bit is self clearing.
Reserved	23:2	0x0	Reserved
Stuck Reason	31:24	0x0	This field can be used by software to indicate to the firmware the reason the 82576 is malfunctioning. The encoding of this field is software/firmware dependent. A value of 0b indicates a functional the 82576.

8.16.3 Free Running Timer - FRTIMER (0x01048; RWS)

This register reflects the value of a free running timer that can be used for various timeout indications. The register is reset by a PCI reset and/or software reset.

Note: Writing to this register is for DFX purposes only.

Field	Bit(s)	Initial Value	Description
Microsecond	9:0	X	Number of microseconds in the current millisecond.
Millisecond	19:10	X	Number of milliseconds in the current second.
Seconds	31:20	X	Number of seconds from the timer start (up to 4095 seconds).

8.16.4 TCP Timer - TCPTIMER (0x0104C; R/W)

Field	Bit(s)	Initial Value	Description
Duration	7:0	0x0	Duration Duration of the TCP interrupt interval in msec.
KickStart (WS)	8	0b	Counter Kick-Start Writing a 1b to this bit kick-starts the counter down-count from the initial value defined in the *Duration* field. Writing a 0b has no effect.
TCPCountEn	9	0b	TCP Count Enable 1b = TCP timer counting enabled. 0b = TCP timer counting disabled. Once enabled, the TCP counter counts from its internal state. If the internal state is equal to 0b, the down-count does not restart until KickStart is activated. If the internal state is not 0b, the down-count continues from internal state. This enables a pause in the counting for debug purpose.

TCPCountFinish (WS)	10	0b	TCP Count Finish This bit enables software to trigger a TCP timer interrupt, regardless of the internal state. Writing a 1b to this bit triggers an interrupt and resets the internal counter to its initial value. Down-count does not restart until either KickStart is activated or Loop is set. Writing a 0b has no effect.
Loop	11	0b	TCP Loop When set to 1b, the TCP counter reloads duration each time it reaches zero, and continues down-counting from this point without kick-starting. When set to 0b, the TCP counter stops at a zero value and does not re-start until KickStart is activated. **Note:** Setting this bit alone is not enough to start the timer activity. The KickStart bit should also be set.
Reserved	31:12	-	Reserved

8.17 Time Sync Register Descriptions

8.17.1 RX Time Sync Control Register - TSYNCRXCTL (0xB620;RW)

Field	Bit(s)	Initial Value	Description
RXTT(RO/V)	0	0x0	Rx timestamp valid (='1' when a valid value for Rx timestamp is captured in the Rx timestamp register, clear by read of Rx timestamp register RXSATRH)
Type	3:1	0x0	Type of packets to timestamp - 000b – time stamp L2 (V2) packets only (Sync or Delay_req depends on message type in Section 8.17.23 and packets with message ID 2 and 3) 001b – time stamp L4 (V1) packets only (Sync or Delay_req depends on message type in Section 8.17.23) 010b – time stamp V2 (L2 and L4) packets (Sync or Delay_req depends on message type in Section 8.17.23 and packets with message ID 2 and 3) 100b – time stamp all packets (in this mode no locking is done to the value in the timestamp registers and no indications in receive descriptors is transferred) 101b - Time stamp all packets which message id bit 3 is zero, which means timestamp all event packets. This is applicable for V2 packets only. 011b, 110b and 111b – reserved
En	4	0b	Enable RX timestamp 0 = time stamping disabled. 1 = time stamping enabled.
Reserved	9:5	0	Reserved
RSV	31:6	0x0	Reserved

8.17.2 RX Timestamp Low - RXSTMPL (0x0B624; RO)

Field	Bit(s)	Initial Value	Description
RXSTMPL	31:0	0x0	Rx timestamp LSB value

8.17.3 RX Timestamp High - RXSTMPH (0x0B628; RO)

Field	Bit(s)	Initial Value	Description
RXSTMPH	31:0	0x0	Rx timestamp MSB value

8.17.4 RX Timestamp Attributes Low - RXSATRL(0x0B62C; RO)

Field	Bit(s)	Initial Value	Description
SourceIDL	31:0	0x0	Sourceuuid low

8.17.5 RX Timestamp Attributes High- RXSATRH (0x0B630; RO)

Field	Bit(s)	Initial Value	Description
SourceIDH	15:0	0x0	Sourceuuid high
SequenceID	31:16	0x0	SequenceId

8.17.6 TX Time Sync Control Register - TSYNCTXCTL (0x0B614; RW)

Field	Bit(s)	Initial Value	Description
TXTT(RO/V)	0	0b	Tx timestamp valid (equals 1b when a valid value for Tx timestamp is captured in the Tx timestamp register, clear by read of Tx timestamp register TXSTMPH)
RSV	3:1	0x0	Reserved
EN	4	0b	Enable TX timestamp 0b = time stamping disabled. 1b = time stamping enabled.
RSV	31:5	0x0	Reserved

8.17.7 TX Timestamp Value Low - TXSTMPL (0x0B618;RO)

Field	Bit(s)	Initial Value	Description
TXSTMPL	31:0	0x0	Tx timestamp LSB value

8.17.8 TX Timestamp Value High - TXSTMPH(0x0B61C; RO)

Field	Bit(s)	Initial Value	Description
TXSTMPH	31:0	0x0	Tx timestamp MSB value

8.17.9 System Time Register Low - SYSTIML (0x0B600; RWS)

Field	Bit(s)	Initial Value	Description
STL	31:0	0x0	system time LSB value

8.17.10 System Time Register High - SYSTIMH (0x0B604; RWS)

Field	Bit(s)	Initial Value	Description
STH	31:0	0x0	system time MSB value

8.17.11 Increment Attributes Register - TIMINCA (0x0B608; RW)

Field	Bit(s)	Initial Value	Description
IV	23:0	0x0	Increment value.
IP	31:24	0x0	Increment period in 16 ns resolution

8.17.12 Time Adjustment Offset Register Low - TIMADJL (0x0B60C; RW)

Field	Bit(s)	Initial Value	Description
TADJL	31:0	0x0	Time adjustment value – Low

8.17.13 Time Adjustment Offset Register High - TIMADJH (0x0B610;RW)

Field	Bit(s)	Initial Value	Description
TADJH	30:0	0x0	Time adjustment value - High
Sign	31	0b	Sign (0b=”+”, 1b =”-“)

8.17.14 TimeSync Auxiliary Control Register - TSAUXC (0x0B640; RW)

Field	Bit(s)	Initial Value	Description
EN_TT0	0	0b	Enable target time 0. Enable bit is set by software to 1b for enabling the feature. The bit is cleared by hardware when the target time is hit.
EN_TT1	1	0b	Enable target time 1. Enable bit is set by software to 1b for enabling the feature. The bit is cleared by hardware when the target time is hit.
reserved	2	0b	Reserved
UTT0	3	0b	Use target time 0 to reload clk_out 0 down counter
ST0	4	0b	Start clock out toggle only on target time0, at this point a rising edge of clock out occurs (the clock output is set to 0b on assertion of this bit).
EN_CLK1	5	0b	Enable Configurable Frequency Clock 1

reserved	5	0b	Reserved
UTT1	6	0b	Use target time 1 to reload clk_out 1 down counter
ST1	7	0b	Start clock out toggle only on target time1, at this point a rising edge of clock out occurs (the clock output is set to 1b on assertion of this bit)
EN_TS0	8	0b	Enable hardware time stamp 0
AUTT0	9	0b	Auxiliary timestamp taken - cleared when read from auxiliary timestamp 0 occurred
EN_TS1	10	0b	Enable hardware time stamp 1
AUTT1	11	0b	Auxiliary timestamp taken - cleared when read from auxiliary timestamp 1 occurred
Mask	16:12	0x0	Masking value for Target Time and Frequency clock accuracy control. The value in this field determines the masked bits in the comparison of the features (where 0b = no masking and 5'd16 the highest value allowed).
RSV	31:17	0x0	Reserved

8.17.15 Target Time Register 0 Low - TRGTTIML0 (0x0B644; RW)

Field	Bit(s)	Initial Value	Description
TTL	31:0	0x0	Target Time 0 LSB register

8.17.16 Target Time Register 0 High - TRGTTIMH0 (0x0B648; RW)

Field	Bit(s)	Initial Value	Description
TTH	31:0	0x0	Target Time 0 MSB register

8.17.17 Target Time Register 1 Low - TRGTTIML1 (0x0B64C; RW)

Field	Bit(s)	Initial Value	Description
TTL	31:0	0x0	Target Time 1 LSB register

8.17.18 Target Time Register 1 High - TRGTTIMH1 (0x0B650; RW)

Field	Bit(s)	Initial Value	Description
TTH	31:0	0x0	Target Time 1 MSB register

8.17.19 Auxiliary Time Stamp 0 Register Low - AUXSTMPL0 (0x0B65C; RO)

Field	Bit(s)	Initial Value	Description
TSTL	31:0	0x0	Auxiliary Time Stamp 0 LSB value

8.17.20 Auxiliary Time Stamp 0 Register High -AUXSTMPH0 (0x0B660; RO)

Reading this register will release the value stored in AUXSTMPH/L0 and will allow stamping of the next value.

Field	Bit(s)	Initial Value	Description
TSTH	31:0	0x0	Auxiliary Time Stamp 0 MSB value

8.17.21 Auxiliary Time Stamp 1 Register Low AUXSTMPL1 (0x0B664; RO)

Field	Bit(s)	Initial Value	Description
TSTL	31:0	0x0	Auxiliary Time Stamp 1 LSB value

8.17.22 Auxiliary Time Stamp 1 Register High - AUXSTMPH1 (0x0B668; RO)

Reading this register will release the value stored in AUXSTMPH/L1 and will allow stamping of the next value.

Field	Bit(s)	Initial Value	Description
TSTH	31:0	0x0	Auxiliary Time Stamp 1 MSB value

8.17.23 Time Sync RX Configuration - TSYNCRXCFG (0x05F50; RW)

Field	Bit(s)	Initial Value	Description
CTRLT	7:0	0x0	V1 control to timestamp
MSGT	11:8	0x0	V2 messageID to timestamp
TRNSSPC	15:12	0x0	V2 value to timestamp
Reserved	31:16	0x0	Reserved

8.17.24 Time Sync SDP Config Reg - TSSDP (0x0003C; RW)

This register defines the assignment of SDP pins to the Time sync auxiliary capabilities.

Field	Bit(s)	Initial Value	Description
AUX0_SPD_SEL	1:0	00b	Select one of the SPDs to serve as the trigger for auxiliary time stamp 0 (AUX0) 00b = SDP0 is assigned 01b = SDP1 is assigned 10b = SDP2 is assigned 11b = SDP3 is assigned
AUX0_TS_SDP_EN	2	0b	When set indicates that one of the SDPs can be used as an external trigger to Aux timestamp 0 (note that if this bit is set to one of the SDP pins, the corresponding pin should be configured to input mode using SPD_DIR)
AUX1_SPD_SEL	4:3	00b	Select one of the SPDs to serve as the trigger for auxiliary time stamp 1 (AUX1) 00b = SDP0 is assigned 01b = SDP1 is assigned 10b = SDP2 is assigned 11b = SDP3 is assigned
AUX1_TS_SDP_EN	5	0b	When set indicates that one of the SDPs can be used as an external trigger to Aux timestamp 1 (note that if this bit is set to one of the SDP pins, the corresponding pin should be configured to input mode using SPD_DIR)
TS_SDP0_SEL	7:6	00b	SDP0 allocation to Tsync event – when TS_SDP0_EN is set, these bits select the Tsync event that is routed to SDP0. 00b = Target Time 0 is output on SDP0 01b = Target Time 1 is output on SDP0 10b - 11b = Reserved
TS_SDP0_EN	8	0b	When set indicates that SDP0 is assigned to Tsync.
TS_SDP1_SEL	10:9	00b	SDP1 allocation to Tsync event – when TS_SDP1_EN is set, these bits select the Tsync event that is routed to SDP1. 00b = Target Time 0 is output on SDP1 01b = Target Time 1 is output on SDP1 10b - 11b = Reserved
TS_SDP1_EN	11	0b	When set indicates that SDP1 is assigned to Tsync.
TS_SDP2_SEL	13:12	00b	SDP2 allocation to Tsync event – when TS_SDP2_EN is set, these bits select the Tsync event that is routed to SDP2. 00b = Target Time 0 is output on SDP2 01b = Target Time 1 is output on SDP2 10b - 11b = Reserved
TS_SDP2_EN	14	0b	When set indicates that SDP2 is assigned to Tsync.
TS_SDP3_SEL	16:15	00b	SDP3 allocation to Tsync event – when TS_SDP3_EN is set, these bits select the Tsync event that is routed to SDP3. 00b = Target Time 0 is output on SDP3 01b = Target Time 1 is output on SDP3 10b - 11b = Reserved
TS_SDP3_EN	17	0b	When set indicates that SDP3 is assigned to Tsync.
Reserved	31:18	0x0	Reserved

8.18 PCS Register Descriptions

The usage of these registers is described in Section 3.5.4.1 & Section 3.5.4.2

8.18.1 PCS Configuration - PCS_CFG (0x04200; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	2:0	000b	Reserved
PCS Enable	3	1b	PCS Enable Enables the PCS logic of the MAC. Should be set in both SGMII and SerDes mode for normal operation. Clearing this bit disables RX/TX of both data and control codes. Use this to force link down at the far end.
Reserved	29:4	0x0	Reserved
PCS Isolate	30	0b	PCS Isolate Setting this bit isolates the PCS logic from the MAC's data path. PCS control codes are still sent and received.
SRESET	31	0b	Soft Reset Setting this bit puts all modules within the MAC in reset except the Host Interface. The Host Interface is reset via HRST. This bit is NOT self clearing; GMAC is in a reset state until this bit is cleared.

8.18.2 PCS Link Control - PCS_LCTL (0x04208; RW)

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved
FSV	2:1	10b	Forced Speed Value These bits denote the speed when force speed and duplex is set. This value is also used when AN is disabled or when in SerDes mode. 00b = 10 Mb/s (SGMII). 01b = 100 Mb/s (SGMII). 10b = 1000 Mb/s (SerDes/SGMII). 11b = Reserved.
FDV	3	1b	Forced Duplex Value This bit denotes the duplex mode when force speed and duplex is set. This value is also used when AN is disabled or when in SerDes mode. 1b = Full duplex (SerDes/SGMII). 0b = Half duplex (SGMII).
FSD	4	0b	Force Speed and Duplex If this bit is set, then speed and duplex mode is forced to forced speed value and forced duplex value, respectively. Otherwise, speed and duplex mode are decided by internal AN/SYNC state machines.
Reserved	5	0b	Reserved - must be set to zero.
LINK LATCH LOW	6	0b	Link Latch Low Enable If this bit is set, then link OK going LOW (negative edge) is latched until a processor read. Afterwards, link OK is continuously updated until link OK again goes LOW (negative edge is seen).
Force Flow Control	7	0b	0 = Flow control mode is set according to the AN process by following Table 37-4 in the IEEE 802.3 spec. 1 = Flow control is set according to FC_TX_EN / FC_RX_EN bits in CTRL register.
Reserved	15:8	-	Reserved
AN_ENABLE	16	0b[1]	AN Enable Setting this bit enables the AN process.

Field	Bit(s)	Initial Value	Description
AN RESTART	17	0b	AN Restart Setting this bit restarts the AN process. This bit is self clearing.
AN TIMEOUT EN	18	1b	AN Timeout Enable This bit enables the AN Timeout feature. During AN, if the link partner does not respond with AN pages, but continues to send good IDLE symbols, then LINK UP is assumed. (This enables LINK UP condition when link partner is not AN-capable and does not affect otherwise). This bit should not be set in SGMII mode.
AN SGMII BYPASS	19	0b	AN SGMII Bypass If this bit is set, then IDLE detect state is bypassed during AN in SGMII mode. This reduces the acknowledge time in SGMII mode.
AN SGMII TRIGGER	20	1b	AN SGMII Trigger If this bit is cleared, then AN is not automatically triggered in SGMII mode even if SYNC fails. AN is triggered only in response to PHY messages or by a manual setting like changing the AN Enable/Restart bits.
Reserved	23:21	000b	Reserved
FAST LINK TIMER	24	0b	Fast Link Timer AN timer is reduced if this bit is set.
LINK OK FIX EN	25	1b	Link OK Fix Enable Control for enabling/disabling LinkOK/SyncOK fix. Should be set for normal operation.
Reserved	26	0b	Reserved
Reserved	31:27	0x0	Reserved

1. Read from EEPROM word 0x0F, bit 11.

8.18.3 PCS Link Status - PCS_LSTS (0x0420C; RO)

Field	Bit(s)	Initial Value	Description
LINK OK	0	0b	Link OK This bit denotes the current link ok status. 0b = Link down. 1b = Link up/OK.
SPEED	2:1	10b	Speed This bit denotes the current operating Speed. 00b = 10 Mb/s. 01b = 100 Mb/s. 10b = 1000 Mb/s. 11b = Reserved.
DUPLEX	3	1b	Duplex This bit denotes the current duplex mode. 1b = Full duplex. 0b = Half duplex.
SYNC OK	4	0b	Sync OK This bit indicates the current value of Sync OK from the PCS Sync state machine.
Reserved	15:5	-	Reserved

Field	Bit(s)	Initial Value	Description
AN COMPLETE	16	0b	AN Complete This bit indicates that the AN process has completed.This bit is set when the AN process reached the Link OK state. It is reset upon AN restart or reset. It is set even if the AN negotiation failed and no common capabilities where found.
AN PAGE RECEIVED	17	0b	AN Page Received This bit indicates that a link partner's page was received during an AN process. This bit is cleared on reads.
AN TIMEDOUT	18	0b	AN Timed Out This bit indicates an AN process was timed out. Valid after the *AN Complete* bit is set.
AN REMOTE FAULT	19	0b	AN Remote Fault This bit indicates that an AN page was received with a remote fault indication during an AN process. This bit cleared on reads.
AN ERROR (RWS)	20	0B	AN Error This bit indicates that a AN error condition was detected in SerDes/SGMII mode. Valid after the *AN Complete* bit is set. AN error conditions: SerDes mode: Both node not Full Duplex SGMII mode: PHY is set to 1000 Mb/s Half Duplex mode. Software can also force a AN error condition by writing to this bit (or can clear a existing AN error condition). This bit is cleared at the start of AN.
Reserved	31:21	0x0	Reserved

8.18.4 AN Advertisement - PCS_ANADV (0x04218; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	4:0	-	Reserved
FDCAP	5	1b	Full Duplex Setting this bit indicates that the 82576 is capable of full duplex operation. This bit should be set to 1b for normal operation.
HDCAP (RO)	6	0b	Half Duplex This bit indicates that the 82576 is capable of half duplex operation. This bit is tied to 0b because the 82576 does not support half duplex in SerDes mode.
ASM	8:7	0b[1]	Local PAUSE Capabilities The 82576's PAUSE capability is encoded in this field. 00b = No PAUSE. 01b = Symmetric PAUSE. 10b = Asymmetric PAUSE to link partner. 11b = Both symmetric and asymmetric PAUSE to the 82576.
Reserved	11:9	-	Reserved
RFLT	13:12	00b	Remote Fault The 82576's remote fault condition is encoded in this field. The 82576 might indicate a fault by setting a non-zero remote fault encoding and re-negotiating. 00b = No error, link OK. 01b = Link failure. 10b = Offline. 11b = Auto-negotiation error.

Field	Bit(s)	Initial Value	Description
Reserved	14	-	Reserved
NEXTP	15	0b	Next Page Capable The 82576 asserts this bit to request a next page transmission. The 82576 clears this bit when no subsequent next pages are requested.
Reserved	31:16	0x0	Reserved

1. Loaded from EEPROM word 0x0F, bits 13:12.

8.18.5 Link Partner Ability - PCS_LPAB (0x0421C; RO)

Field	Bit(s)	Initial Value	Description
Reserved	4:0	-	Reserved
LPFD	5	0b	LP Full Duplex (SerDes) When set to 1b, the link partner is capable of full duplex operation. When set to 0b, the link partner is not capable of full duplex mode. This bit is reserved while in SGMII mode.
LPHD	6	0b	LP Half Duplex (SerDes) When set to 1b, the link partner is capable of half duplex operation. When set to 0b, the link partner is not capable of half duplex mode. This bit is reserved while in SGMII mode.
LPASM	8:7	00b	LP ASMDR/LP PAUSE (SerDes) The link partner's PAUSE capability is encoded in this field. 00b = No PAUSE. 01b = Symmetric PAUSE. 10b = Asymmetric PAUSE to link partner. 11b = Both symmetric and asymmetric PAUSE to the 82576. These bits are reserved while in SGMII mode.
Reserved	9	-	Reserved
SGMII SPEED	11:10	00b	SerDes: reserved. Speed (SGMII): Speed indication from the PHY.
PRF	13:12	00b	LP Remote Fault (SerDes) The link partner's remote fault condition is encoded in this field. 00b = No error, link ok. 10b = Link failure. 01b = Offline. 11b = Auto-negotiation error. SGMII[13]: Reserved SGMII[12]: Duplex mode indication from the PHY.
ACK	14	0b	Acknowledge (SerDes) The link partner has acknowledge page reception. SGMII: Reserved.
LPNEXTP	15	0b	LP Next Page Capable (SerDes) The link partner asserts this bit to indicate its ability to accept next pages. SGMII: Link-OK indication from the PHY.
Reserved	31:16	-	Reserved

8.18.6 Next Page Transmit - PCS_NPTX (0x04220; RW)

Field	Bit(s)	Initial Value	Description
CODE	10:0	0x0	Message/Unformatted Code Field The Message Field is an 11-bit wide field that encodes 2048 possible messages. Unformatted Code Field is an 11-bit wide field that might contain an arbitrary value.
TOGGLE	11	0b	Toggle This bit is used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes the opposite value of the *Toggle* bit in the previously exchanged Link Code Word. The initial value of the *Toggle* bit in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word and, therefore, can assume a value of 0b or 1b. The *Toggle* bit is set as follows: 0b = Previous value of the transmitted Link Code Word when 1b 1b = Previous value of the transmitted Link Code Word when 0b.
ACK2	12	0b	Acknowledge 2 Used to indicate that a device has successfully received its Link Partners' Link Code Word.
PGTYPE	13	0b	Message/Unformatted Page This bit is used to differentiate a Message Page from an Unformatted Page. The encoding is: 0b = Unformatted page. 1b = Message page.
Reserved	14	-	Reserved
NXTPG	15	0b	Next Page Used to indicate whether or not this is the last Next Page to be transmitted. The encoding is: 0b = Last page. 1b = Additional Next Pages follow.
Reserved	31:16	-	Reserved

8.18.7 Link Partner Ability Next Page - PCS_LPABNP (0x04224; RO)

Field	Bit(s)	Initial Value	Description
CODE	10:0	-	Message/Unformatted Code Field The Message Field is an 11-bit wide field that encodes 2048 possible messages. Unformatted Code Field is an 11-bit wide field that might contain an arbitrary value.
TOGGLE	11	-	Toggle This bit is used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes the opposite value of the *Toggle* bit in the previously exchanged Link Code Word. The initial value of the *Toggle* bit in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word and, therefore, can assume a value of 0b or 1b. The *Toggle* bit is set as follows: 0b = Previous value of the transmitted Link Code Word when 1b 1b = Previous value of the transmitted Link Code Word when 0b.
ACK2	12	-	Acknowledge 2 Used to indicate that a device has successfully received its Link Partners' Link Code Word.

Field	Bit(s)	Initial Value	Description
MSGPG	13	-	Message Page This bit is used to differentiate a Message Page from an Unformatted Page. The encoding is: 0b = Unformatted page. 1b = Message page.
ACK	14	-	Acknowledge The Link Partner has acknowledged Next Page reception.
NXTPG	15	-	Next Page Used to indicate whether or not this is the last Next Page to be transmitted. The encoding is: 0b = Last page. 1b = Additional Next Pages follow.
Reserved	31:16	-	Reserved

8.18.8 SFP I2C Command- I2CCMD (0x01028; R/W)

This register is used by software to read or write to the configuration registers in an SFP module.

Note: According to the SFP specification, only reads are allowed from this interface; however, SFP vendors also provide a writable register through this interface (for example, PHY registers). As a result, write capability is also supported.

Field	Bit(s)	Initial Value	Description
DATA	15:0	X	Data In a write command, software places the data bits and then the MAC shifts them out to the I^{2}C bus. In a read command, the MAC reads these bits serially from the I^{2}C bus and then software reads them from this location. **Note: This field is read in byte order not in word order.**
REGADD	23:16	0x0	I^{2}C Register Address For example, register 0, 1, 2, . . . 255.
PHYADD	26:24	0x0	Device Address bits 3 -1 The actual address used is b{1010, PHYADD[2:0], 0}.
OP	27	0b	Op Code 0b = I^{2}C write. 1b = I^{2}C read.
Reset	28	0b	Reset Sequence If set, sends a reset sequence before the actual read or write. This bit is self clearing. A reset sequence is defined as nine consecutive stop conditions.
R	29	0b	Ready Bit Set to 1b by the 82576 at the end of the I^{2}C transaction. For example, indicates a read or write has completed. Reset by a software write of a command.
Reserved	30	0b	Reserved
E	31	0b	Error This bit set is to 1b by hardware when it fails to complete an I^{2}C read. Reset by a software write of a command.

8.18.9 SFP I2C Parameters - I2CPARAMS (0x0102C; R/W)

This register is used to set the parameters for the I^{2}C access to the SFP module and to allow bit bang access to the I^{2}C interface.

Field	Bit(s)	Initial Value	Description
Write Time	4:0	110b	Write Time Defines the delay between a write access and the next access. The value is in milliseconds. A value of zero is not valid.
Read Time	7:5	010b	Read Time Defines the delay between a read access and the next access. The value is in microseconds. A value of Zero is not valid
I2CBB_EN	8	0b	I^{2}C Bit Bang Enable If set, the I^{2}C_CLK and I^{2}C_DATA lines are controlled via the CLK, DATA and DATA_OE_N fields of this register. Otherwise, they are controlled by the hardware machine activated via the I2CCMD or MDIC registers.
CLK	9	0b	I^{2}C Clock While in bit bang mode, controls the value driven on the I2C_CLK pad of this port.
DATA_OUT	10	0b	I^{2}C_DATA While in bit bang mode and when the DATA_OE_N field is zero, controls the value driven on the I2C_DATA pad of this port.
DATA_OE_N	11	0b	I^{2}C_DATA_OE_N While in bit bang mode, controls the direction of the I2C_DATA pad of this port. 0b = Pad is output. 1b = Pad is input.
DATA_IN (RO)	12	X	I^{2}C_DATA_IN Reflects the value of the I2C_DATA pad. While in bit bang mode and when the DATA_OE_N field is zero, this field reflects the value set in the DATA_OUT field.
Reserved	31:13	0x0	Reserved

8.19 Statistics Register Descriptions

All Statistics registers reset when read. In addition, they stick at 0xFFFF_FFFF when the maximum value is reached.

For the receive statistics it should be noted that a packet is indicated as received if it passes the 82576's filters and is placed into the packet buffer memory. A packet does not have to be transferred to host memory in order to be counted as received.

Due to divergent paths between interrupt-generation and logging of relevant statistics counts, it might be possible to generate an interrupt to the system for a noteworthy event prior to the associated statistics count actually being incremented. This is extremely unlikely due to expected delays associated with the system interrupt-collection and ISR delay, but might be observed as an interrupt for which statistics values do not quite make sense. Hardware guarantees that any event noteworthy of inclusion in a statistics count is reflected in the appropriate count within 1 µs; a small time-delay prior to a read of statistics might be necessary to avoid the potential for receiving an interrupt and observing an inconsistent statistics count as part of the ISR.

8.19.1 CRC Error Count - CRCERRS (0x04000; 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.

Field	Bit(s)	Initial Value	Description
CEC	31:0	0x0	CRC error count

8.19.2 Alignment Error Count - ALGNERRC (0x04004; 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>, inclusive) 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.

Field	Bit(s)	Initial Value	Description
AEC	31:0	0x0	Alignment error count

8.19.3 Symbol Error Count - SYMERRS (0x04008; 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. When working in SerDes/SGMII mode these statistics can be read from the SCVPC register.

Field	Bit(s)	Initial Value	Description
SYMERRS	31:0	0x0	Symbol Error Count

8.19.4 RX Error Count - RXERRC (0x0400C; RC)

Counts the number of packets received in which 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>, inclusive) in length. If receives are not enabled, then this register does not increment.

This register is not available in SerDes/SGMII modes.

Field	Bit(s)	Initial Value	Description
RXEC	31:0	0x0	RX error count

8.19.5 Missed Packets Count - MPC (0x04010; 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.

Field	Bit(s)	Initial Value	Description
MPC	31:0	0x0	Missed Packets Count

Table 8-22. Single Collision Count - SCC (0x04014; 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 82576 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
SCC	31:0	0x0	Number of times a transmit encountered a single collision.

8.19.6 Excessive Collisions Count - ECOL (0x04018; 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 82576 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
ECC	31:0	0x0	Number of packets with more than 16 collisions.

8.19.7 Multiple Collision Count - MCC (0x0401C; 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 82576 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
MCC	31:0	0x0	Number of times a successful transmit encountered multiple collisions.

8.19.8 Late Collisions Count - LATECOL (0x04020; RC)

Late collisions are collisions that occur after one slot time. This register only increments if transmits are enabled and the 82576 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
LCC	31:0	0x0	Number of packets with late collisions.

8.19.9 Collision Count - COLC (0x04028; RC)

This register counts the total number of collisions seen by the transmitter. This register only increments if transmits are enabled and the 82576 is in half-duplex mode. This register applies to clear as well as secure traffic.

Field	Bit(s)	Initial Value	Description
CCC	31:0	0x0	Total number of collisions experienced by the transmitter.

8.19.10 Defer Count - DC (0x04030; 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.

Field	Bit(s)	Initial Value	Description
CDC	31:0	0x0	Number of defer events.

8.19.11 Transmit with No CRS - TNCRS (0x04034; RC)

This register counts the number of successful packet transmissions in which the CRS input from the PHY was not asserted within one slot time of start of transmission from the MAC. Start of transmission is defined as the assertion of TX_EN to the PHY.

The PHY should assert CRS during every transmission. Failure to do so might 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 SGMII mode and is only valid when the 82576 is operating at half duplex.

Field	Bit(s)	Initial Value	Description
TNCRS	31:0	0x0	Number of transmissions without a CRS assertion from the PHY.

8.19.12 Host Transmit Discarded Packets by MAC Count - HTDPMC (0x0403C; RC)

Field	Bit(s)	Initial Value	Description
HTDPMC	31:0	0x0	Number of packets sent by the host but discarded by the MAC

This register counts the number of packets sent by the host (and not the manageability engine) that are dropped by the MAC. This can include packets dropped because of excessive collisions or link fail events.

8.19.13 Receive Length Error Count - RLEC (0x04040; 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 1518/1522/1526 bytes (according to the number of VLAN tags present) are oversized if Long Packet Enable (LPE) is 0b. If LPE is 1b, then an incoming, packet is considered oversized if it exceeds the size defined in RLPML.RLPML field.

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>, inclusive.

Note: Runt packets smaller than 25 bytes may not be counted by this counter.

Field	Bit(s)	Initial Value	Description
RLEC	31:0	0x0	Number of packets with receive length errors.

8.19.14 XON Received Count - XONRXC (0x04048; 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.

Field	Bit(s)	Initial Value	Description
XONRXC	31:0	0x0	Number of XON packets received.

8.19.15 XON Transmitted Count - XONTXC (0x0404C; 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.

Field	Bit(s)	Initial Value	Description
XONTXC	31:0	0x0	Number of XON packets transmitted.

8.19.16 XOFF Received Count - XOFFRXC (0x04050; 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.

Field	Bit(s)	Initial Value	Description
XOFFRXC	31:0	0x0	Number of XOFF packets received.

8.19.17 XOFF Transmitted Count - XOFFTXC (0x04054; 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.

Field	Bit(s)	Initial Value	Description
XOFFTXC	31:0	0x0	Number of XOFF packets transmitted.

8.19.18 FC Received Unsupported Count - FCRUC (0x04058; 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.

Field	Bit(s)	Initial Value	Description
FCRUC	31:0	0x0	Number of unsupported flow control frames received.

8.19.19 Packets Received [64 Bytes] Count - PRC64 (0x0405C; RC)

This register counts the number of good packets received that are exactly 64 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled.

Field	Bit(s)	Initial Value	Description
PRC64	31:0	0x0	Number of packets received that are 64 bytes in length.

8.19.20 Packets Received [65—127 Bytes] Count - PRC127 (0x04060; RC)

This register counts the number of good packets received that are 65-127 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled.

Field	Bit(s)	Initial Value	Description
PRC127	31:0	0x0	Number of packets received that are 65-127 bytes in length.

8.19.21 Packets Received [128—255 Bytes] Count - PRC255 (0x04064; RC)

This register counts the number of good packets received that are 128-255 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled.

Field	Bit(s)	Initial Value	Description
PRC255	31:0	0x0	Number of packets received that are 128-255 bytes in length.

8.19.22 Packets Received [256—511 Bytes] Count - PRC511 (0x04068; RC)

This register counts the number of good packets received that are 256-511 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled.

Field	Bit(s)	Initial Value	Description
PRC511	31:0	0x0	Number of packets received that are 256-511 bytes in length.

8.19.23 Packets Received [512—1023 Bytes] Count - PRC1023 (0x0406C; RC)

This register counts the number of good packets received that are 512-1023 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled.

Field	Bit(s)	Initial Value	Description
PRC1023	31:0	0x0	Number of packets received that are 512-1023 bytes in length.

8.19.24 Packets Received [1024 to Max Bytes] Count - PRC1522 (0x04070; RC)

This register counts the number of good packets received that are from 1024 bytes to the maximum (from <Destination Address> through <CRC>, inclusive) in length. The maximum is dependent on the current receiver configuration (for example, LPE, etc.) 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 8.19.36). This register does not include received flow control packets and only increments if the packet has passed address filtering and receives are enabled. Packets sent to the manageability engine are included in this counter.

Due to changes in the standard for maximum frame size for VLAN tagged frames in 802.3, the 82576 accepts 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. If CTRL.Extended_VLAN is set, packets up to 1526 bytes are counted by this counter.

Field	Bit(s)	Initial Value	Description
PRC1522	31:0	0x0	Number of packets received that are 1024-Max bytes in length.

8.19.25 Good Packets Received Count - GPRC (0x04074; 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 Long Packet Enable (CTRL.LPE) (see Section 8.19.36). 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. Packets sent to the manageability engine are included in this counter.

Note: GPRC can count packets interrupted by a link disconnect although they have a CRC error.

Field	Bit(s)	Initial Value	Description
GPRC	31:0	0x0	Number of good packets received (of any length).

8.19.26 Broadcast Packets Received Count - BPRC (0x04078; 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. Packets sent to the manageability engine are included in this counter.

Field	Bit(s)	Initial Value	Description
BPRC	31:0	0x0	Number of broadcast packets received.

8.19.27 Multicast Packets Received Count - MPRC (0x0407C; 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. Packets sent to the manageability engine are included in this counter.

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0x0	Number of multicast packets received.

8.19.28 Good Packets Transmitted Count - GPTC (0x04080; 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. The register counts clear as well as secure packets.

Field	Bit(s)	Initial Value	Description
GPTC	31:0	0x0	Number of good packets transmitted.

8.19.29 Good Octets Received Count - GORCL (0x04088; 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, inclusive; GORCL must be read before GORCH.

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. Only octets of packets that pass address filtering are counted in this register. This register does not count octets of packets counted by the Missed Packet Count (MPC) register. Octets of packets sent to the manageability engine are included in this counter. This register only increments if receives are enabled.

These octets do not include octets of received flow control packets.

Field	Bit(s)	Initial Value	Description
GORCL	31:0	0x0	Number of good octets received – lower 4 bytes.

8.19.30 Good Octets Received Count - GORCH (0x0408C; RC)

Field	Bit(s)	Initial Value	Description
GORCH	31:0	0x0	Number of good octets received – upper 4 bytes.

8.19.31 Good Octets Transmitted Count - GOTCL (0x04090; RC)

These registers make up a 64-bit register that counts the number of good (no errors) packets transmitted. This register must be accessed using two independent 32-bit accesses; GOTCL must be read before GOTCH.

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register counts octets in successfully transmitted packets that are 64 or more bytes in length. This register only increments if transmits are enabled. The register counts clear as well as secure octets.

These octets do not include octets in transmitted flow control packets.

Field	Bit(s)	Initial Value	Description
GOTCL	31:0	0x0	Number of good octets transmitted – lower 4 bytes.

8.19.32 Good Octets Transmitted Count - GOTCH (04094; RC)

Field	Bit(s)	Initial Value	Description
GOTCH	31:0	0x0	Number of good octets transmitted – upper 4 bytes.

8.19.33 Receive No Buffers Count - RNBC (0x040A0; 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.

Field	Bit(s)	Initial Value	Description
RNBC	31:0	0x0	Number of receive no buffer conditions.

8.19.34 Receive Undersize Count - RUC (0x040A4; 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>, inclusive), and had a valid CRC. This register only increments if receives are enabled.

Field	Bit(s)	Initial Value	Description
RUC	31:0	0x0	Number of receive undersize errors.

8.19.35 Receive Fragment Count - RFC (0x040A8; 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>, inclusive), but had a bad CRC (this is slightly different from the Receive Undersize Count register). This register only increments if receives are enabled.

Field	Bit(s)	Initial Value	Description
RFC	31:0	0x0	Number of receive fragment errors.

Note: Runt packets smaller than 25 bytes may not be counted by this counter.

8.19.36 Receive Oversize Count - ROC (0x040AC; 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 the value set in the RLPML register.

In Next Generation VMDq mode, a packet is counted only if it is bigger than the VOMLR.RLPML value for all the VFs that where supposed to receive the packet.

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>, inclusive.

Note: The maximum size of a packet when LPE is 0b is fixed according to the CTRL_EXT.Extended_VLAN bit and the detection of a VLAN tag in the packet.

Field	Bit(s)	Initial Value	Description
ROC	31:0	0x0	Number of receive oversize errors.

8.19.37 Receive Jabber Count - RJC (0x040B0; 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 1518/1522/1526 bytes are oversized if LPE is 0b. If LPE is 1b, then an incoming packet is considered oversized if it exceeds RLPML.LPML 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>, inclusive.

Note: The maximum size of a packet when LPE is 0b is fixed according to the CTRL_EXT.Extended_VLAN bit and the detection of a VLAN tag in the packet.

Field	Bit(s)	Initial Value	Description
RJC	31:0	0x0	Number of receive jabber errors.

8.19.38 Management Packets Received Count - MNGPRC (0x040B4; RC)

This register counts the total number of packets received that pass the management filters as described in the Total Cost of Ownership (TCO) System Management Bus Interface Application Note. Any packets with errors are not counted, except packets that are dropped because the management receive FIFO is full.

Packets sent to both the host and the management interface are not counted by this counter.

Field	Bit(s)	Initial Value	Description
MNGPRC	31:0	0x0	Number of management packets received.

8.19.39 BMC Management Packets Received Count - BMNGPRC (0x0413C; RC)

This register counts the total number of packets received that pass the management filters as described in the Total Cost of Ownership (TCO) System Management Bus Interface Application Note. Any packets with errors are not counted, except packets that are dropped because the management receive FIFO is full.

This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
MNGPRC	31:0	0x0	Number of management packets received.

8.19.40 Management Packets Dropped Count - MPDC (0x040B8; RC)

This register counts the total number of packets received that pass the management filters as described in the Total Cost of Ownership (TCO) System Management Bus Interface Application Note, that are dropped because the management receive FIFO is full. Management packets include any packet directed to the manageability console (for example, BMC and ARP packets).

Field	Bit(s)	Initial Value	Description
MPDC	31:0	0x0	Number of management packets dropped.

8.19.41 BMC Management Packets Dropped Count - BMPDC (0x04140; RC)

This register counts the total number of packets received that pass the management filters as described in the Total Cost of Ownership (TCO) System Management Bus Interface Application Note, that are dropped because the management receive FIFO is full. Management packets include any packet directed to the manageability console (for example, BMC and ARP packets).

This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
MPDC	31:0	0x0	Number of management packets dropped.

8.19.42 Management Packets Transmitted Count - MNGPTC (0x040BC; RC)

This register counts the total number of transmitted packets originating from the manageability path.

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0x0	Number of management packets transmitted.

8.19.43 BMC Management Packets Transmitted Count - BMNGPTC (0x04144; RC)

This register counts the total number of transmitted packets originating from the manageability path.

This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0x0	Number of management packets transmitted.

8.19.44 Total Octets Received - TORL (0x040C0; RC)

These registers make up a logical 64-bit register which counts the total number of octets received. This register must be accessed using two independent 32-bit accesses; TORL must be read before TORH. This register sticks at 0xFFFF_FFFF_FFFF_FFFF 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, inclusive. This register only increments if receives are enabled.

Note: Broadcast rejected packets are counted in this counter (as opposed to all other rejected packets that are not counted).

Field	Bit(s)	Initial Value	Description
TORL	31:0	0x0	Number of total octets received – lower 4 bytes.

8.19.45 Total Octets Received - TORH (0x040C4; RC)

Field	Bit(s)	Initial Value	Description
TORH	31:0	0x0	Number of total octets received – upper 4 bytes.

8.19.46 Total Octets Transmitted - TOTL (0x040C8; RC)

These registers make up a 64-bit register that counts the total number of octets transmitted. This register must be accessed using two independent 32-bit accesses; TOTL must be read before TOTH. This register sticks at 0xFFFF_FFFF_FFFF_FFFF 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, inclusive.

Octets transmitted as part of partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled.

Field	Bit(s)	Initial Value	Description
TOTL	31:0	0x0	Number of total octets transmitted – lower 4 bytes.

8.19.47 Total Octets Transmitted - TOTH (0x040CC; RC)

Field	Bit(s)	Initial Value	Description
TOTH	31:0	0x0	Number of total octets transmitted – upper 4 bytes.

8.19.48 Total Packets Received - TPR (0x040D0; 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.

Note: Broadcast rejected packets are counted in this counter (as opposed to all other rejected packets that are not counted).

Runt packets smaller than 25 bytes may not be counted by this counter.

TPR can count packets interrupted by a link disconnect although they have a CRC error.

Field	Bit(s)	Initial Value	Description
TPR	31:0	0x0	Number of all packets received.

8.19.49 Total Packets Transmitted - TPT (0x040D4; 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 the PT function.

Field	Bit(s)	Initial Value	Description
TPT	31:0	0x0	Number of all packets transmitted.

8.19.50 Packets Transmitted [64 Bytes] Count - PTC64 (0x040D8; RC)

This register counts the number of packets transmitted that are exactly 64 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register does not include transmitted flow control packets (which are 64 bytes in length). This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
PTC64	31:0	0x0	Number of packets transmitted that are 64 bytes in length.

8.19.51 Packets Transmitted [65—127 Bytes] Count - PTC127 (0x040DC; RC)

This register counts the number of packets transmitted that are 65-127 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
PTC127	31:0	0x0	Number of packets transmitted that are 65-127 bytes in length.

8.19.52 Packets Transmitted [128—255 Bytes] Count - PTC255 (0x040E0; RC)

This register counts the number of packets transmitted that are 128-255 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
PTC255	31:0	0x0	Number of packets transmitted that are 128-255 bytes in length.

8.19.53 Packets Transmitted [256—511 Bytes] Count - PTC511 (0x040E4; RC)

This register counts the number of packets transmitted that are 256-511 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, 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 and secure packets. Management packets must never be more than 200 bytes.

Field	Bit(s)	Initial Value	Description
PTC511	31:0	0x0	Number of packets transmitted that are 256-511 bytes in length.

8.19.54 Packets Transmitted [512—1023 Bytes] Count - PTC1023 (0x040E8; RC)

This register counts the number of packets transmitted that are 512-1023 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, 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 and secure packets. Management packets must never be more than 200 bytes.

Field	Bit(s)	Initial Value	Description
PTC1023	31:0	0x0	Number of packets transmitted that are 512-1023 bytes in length.

8.19.55 Packets Transmitted [1024 Bytes or Greater] Count - PTC1522 (0x040EC; RC)

This register counts the number of packets transmitted that are 1024 or more bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled.

Due to changes in the standard for maximum frame size for VLAN tagged frames in 802.3, the 82576 transmits 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 and secure packets (management packets must never be more than 200 bytes). If CTRL.Extended_VLAN is set, packets up to 1526 bytes are counted by this counter.

Field	Bit(s)	Initial Value	Description
PTC1522	31:0	0x0	Number of packets transmitted that are 1024 or more bytes in length.

8.19.56 Multicast Packets Transmitted Count - MPTC (0x040F0; 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.

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0x0	Number of multicast packets transmitted.

8.19.57 Broadcast Packets Transmitted Count - BPTC (0x040F4; RC)

This register counts the number of broadcast packets transmitted. This register only increments if transmits are enabled. This register counts all packets, including standard and secure packets (management packets must never be more than 200 bytes).

Field	Bit(s)	Initial Value	Description
BPTC	31:0	0x0	Number of broadcast packets transmitted count.

8.19.58 TCP Segmentation Context Transmitted Count - TSCTC (0x040F8; 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 on-chip transmit buffer. Note that it 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.

This counter only counts pure TSO transmissions.

Field	Bit(s)	Initial Value	Description
TSCTC	31:0	0x0	Number of TCP Segmentation contexts transmitted count.

8.19.59 Interrupt Assertion Count - IAC (0x04100; RC)

This counter counts the total number of LAN interrupts generated in the system. In case of MSI-X systems, this counter reflects the total number of MSI-X messages that are emitted.

Field	Bit(s)	Initial Value	Description
IAC	31:0	0x0	This is a count of all the LAN interrupt assertions that have occurred.

8.19.60 Rx Packets to Host Count - RPTHC (0x04104; RC)

Field	Bit(s)	Initial Value	Description
RPTHC	31:0	0x0	This is a count of all the received packets sent to the host.

8.19.61 Debug Counter 1 - DBGC1 (0x04108; RC)

Field	Bit(s)	Initial Value	Description
DBGC1	31:0	0x0	This field counts the events according to the value of the PBDIAG.STAT_SEL field. The list of possible values for this counter are described in Table 8-23:

Table 8-23. DBGC1 Values

Stat Sel	Counter 1 content
0	Number of Switch packets read from DBU0
1	The number of Tx descriptor WB transactions performed for Q0
2	The number of Rx descriptor WB transactions performed for Q0
3	The number of Rx descriptor immediate WB transactions performed for Q0
4	The number of Tx host descriptors read by the Descriptor Handler
5	The number of Rx host descriptors read by the Descriptor Processor
6	The number of Tx data read requests done by the Dhost
7	The number of TX packets sent to DBU

8.19.62 Debug Counter 2 - DBGC2 (0x0410C; RC)

Field	Bit(s)	Initial Value	Description
DBGC2	31:0	0x0	This field counts the events according to the value of the PBDIAG.STAT_SEL field. The list of possible values for this counter are described in Table 8-24.

Table 8-24. DBGC2 Values

Stat sel	Counter 2 content
0	Number of Rx filter packets read from DBU0
1	The number of Tx descriptor WB transactions performed for Q1
2	The number of Rx descriptor WB transactions performed for Q1
3	The number of Rx descriptor immediate WB transactions performed for Q1
4	The number of Tx host descriptors written back to host
5	The number of Rx host descriptors written back to host
6	The number of Rx data write requests done by the Dhost
7	Reserved

8.19.63 Debug Counter 3 - DBGC3 (0x04110; RC)

Field	Bit(s)	Initial Value	Description
DBGC3	31:0	0x0	This field counts the events according to the value of the PBDIAG.STAT_SEL field. The list of possible values for this counter are described in Table 8-25:

Table 8-25. DBGC3 Values

Stat sel	Counter 3 content
0	Number of Switch packets read from DBU1
1	The number of Tx descriptor WB transactions performed for Q2
2	The number of Rx descriptor WB transactions performed for Q2
3	The number of Rx descriptor immediate WB transactions performed for Q2
4	The number of dropped Tx packets
5	The number of Rx packets written to the DBU
6	The number of PCI write requests done by the DMA
7	The total amount of single send packets

8.19.64 Debug Counter 4 - DBGC4 (0x0411C; RC)

Field	Bit(s)	Initial Value	Description
DBGC4	31:0	0x0	This field counts the events according to the value of the PBDIAG.STAT_SEL field. The list of possible values for this counter are described in Table 8-26:

Table 8-26. DBGC4 Values

Stat sel	Counter 4 content
0	Number of Rx filter packets read from DBU1
1	The number of Tx descriptor WB transactions performed for Q3
2	The number of Rx descriptor WB transactions performed for Q3
3	The number of Rx descriptor immediate WB transactions performed for Q3
4	The number of Tx packets read from the DBU
5	The number of Rx packets read from the DBU
6	Number of PCI read/write requests done by the Dhost
7	The total amount of large send packets

8.19.65 Host Good Packets Transmitted Count-HGPTC (0x04118; RC)

Field	Bit(s)	Initial Value	Description
HGPTC	31:0	0x0	Number of good packets transmitted by the host.

This register counts the number of good (non-erred) packets transmitted sent by the host. 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 or packets sent by the manageability engine. This register only increments if transmits are enabled.

8.19.66 Receive Descriptor Minimum Threshold Count-RXDMTC (0x04120; RC)

Field	Bit(s)	Initial Value	Description
RXDMTC	31:0	0x0	This is a count of the receive descriptor minimum threshold events

This register counts the number of events where the number of descriptors in one of the Rx queues was lower than the threshold defined for this queue.

8.19.67 Host Good Octets Received Count - HGORCL (0x04128; RC)

Field	Bit(s)	Initial Value	Description
HGORCL	31:0	0x0	Number of good octets received by host – lower 4 bytes

8.19.68 Host Good Octets Received Count - HGORCH (0x0412C; RC)

Field	Bit(s)	Initial Value	Description
HGORCH	31:0	0x0	Number of good octets received by host – upper 4 bytes

These registers make up a logical 64-bit register which counts the number of good (non-erred) octets received. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register must be accessed using two independent 32-bit accesses.; HGORCL must be read before HGORCH.

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. Only packets that pass address filtering are counted in this register. This register counts only octets of packets that reached the host. The only exception is packets dropped by the DMA because of lack of descriptors in one of the queues. These packets are included in this counter.

This register only increments if receives are enabled.

8.19.69 Host Good Octets Transmitted Count - HGOTCL (0x04130; RC)

Field	Bit(s)	Initial Value	Description
HGOTCL	31:0	0x0	Number of good octets transmitted by host – lower 4 bytes

8.19.70 Host Good Octets Transmitted Count - HGOTCH (0x04134; RC)

Field	Bit(s)	Initial Value	Description
HGOTCH	31:0	0x0	Number of good octets transmitted by host – upper 4 bytes

These registers make up a logical 64-bit register which counts the number of good (non-erred) packets transmitted. This register must be accessed using two independent 32-bit accesses. This register resets whenever the upper 32 bits are read (HGOTCH).

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register counts octets in successfully transmitted packets which are 64 or more bytes in length. This register only increments if transmits are enabled. The register counts clear as well as secure octets.

These octets do not include octets in transmitted flow control packets or manageability packets.

8.19.71 Length Error Count - LENERRS (0x04138; RC)

Field	Bit(s)	Initial Value	Description
LENERRS	31:0	0x0	Length error count.

Counts the number of receive packets with Length errors. For example, valid packets (no CRC error) with a length/Type field with a value smaller or equal to 1500 greater than the frame size. 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>, inclusive) in length. If receives are not enabled, then this register does not increment.

8.19.72 SerDes/SGMII Code Violation Packet Count - SCVPC (0x04228; RW)

This register contains the number of code violation packets received. Code violation is defined as an invalid received code in the middle of a packet.

Field	Bit(s)	Initial Value	Description
CODEVIO	31:0	0x0	Code Violation Packet Count: At any point of time this field specifies number of unknown protocol packets received. Valid only in SGMII/SerDes mode.

8.19.73 Switch Security Violation Packet Count - SSVPC (0x41A0; RC)

This register counts Tx packets dropped due to switch security violations such as an SA anti spoof filtering. Valid only in Next Generation VMDq or IOV mode.

Field	Bit(s)	Initial Value	Description
SSVPC	31:0	0x0	Switch Security Violation Packet Count: This register counts Tx packets dropped due to switch security violations such as an SA anti spoof filtering.

8.19.74 Switch Drop Packet Count - SDPC (0x41A4; RC)

Field	Bit(s)	Initial Value	Description
SDPC	31:0	0x0	Switch Drop Packet Count: This register counts Rx packets dropped at the pool selection stage of the switch or by the storm control mechanism. For example, packets that where not routed to any of the pools and the VT_CTL.Dis_Def_Pool is set. Valid only in Next Generation VMDq or IOV mode.

8.20 Wake Up Control Register Descriptions

8.20.1 Wakeup Control Register - WUC (0x05800; R/W)

The PME_En and PME_Status bits of this register are reset on Internal_Power_On_Reset event. When AUX_PWR = 0b, this register is also reset by de-asserting PE_RST_N and during a D3 to D0 transition. The other bits are reset using the standard internal resets.

Field	Bit(s)	Initial Value	Description
APME	0	0b[1]	Advance Power Management Enable If set to 1b, APM Wakeup is enabled. If this bit is set and the *APMPME* bit is cleared, reception of a magic packet asserts the *WUS.MAG* bit but does not assert a PME.
PME_En	1	0b	PME_En This read/write bit is used by the software device driver to access the PME_En bit of the Power Management Control / Status Register (PMCSR) without writing to the PCIe configuration space.

Field	Bit(s)	Initial Value	Description
PME_Status	2	0b	PME_Status This bit is set when the 82576 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[1]	Assert PME On APM Wakeup If set to 1b, the 82576 sets the PME_Status bit in the Power Management Control / Status Register (PMCSR) and asserts PME# when APM Wakeup is enabled and the 82576 receives a matching Magic Packet.
Reserved	31:4	0x0	Reserved

[1] Loaded from the EEPROM.

8.20.2 Wakeup Filter Control Register - WUFC (0x05808; 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.; A value of 0b means the filter is turned off.

If the NoTCO bit is set, then any packet that passes the manageability packet filtering described in the Total Cost of Ownership (TCO) System Management Bus Interface Application Note does not cause a Wake Up event even if it passes one of the Wake Up Filters. This bit is set at initialization and during any EEPROM read if the SMBus Enable bit of the EEPROM's Management Control word is 1b. Otherwise its initial value is 0b.

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.[1]
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.
FLX1	17	0b	Flexible Filter 1 Enable.
FLX2	18	0b	Flexible Filter 2 Enable.
FLX3	19	0b	Flexible Filter 3 Enable.
FLX4	20	0b	Flexible Filter 4 Enable.
FLX5	21	0b	Flexible Filter 5 Enable.
Reserved	31:22	0x0	Reserved.

[1] If the RCTL.UPE is set, and the EX bit is set also, any unicast packet wakes up the system.

8.20.3 Wakeup Status Register - WUS (0x05810; R/W1C)

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 Internal_Power_On_Reset is de-asserted or when cleared by the software device driver.

Note: If additional packets are received that matches one of the wakeup filters, after the original wakeup packet is received, the WUS register is updated with the matching filters accordingly.

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 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.
MNG	8	0b	Indicates that a manageability event that should cause a PME happened.
Reserved	15:9	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.
FLX4	20	0b	Flexible Filter 4 Match.
FLX5	21	0b	Flexible Filter 5 Match.
Reserved	31:22	0b	Reserved.

8.20.4 Wakeup Packet Length - WUPL (0x05900; RO)

This register indicates the length of the first wakeup packet received. It is valid if one of the bits in the Wakeup Status register (WUS) is set. It is not cleared by any reset.

Field	Bit(s)	Initial Value	Description
LEN	11:0	X	Length of wakeup packet. (If jumbo frames is enabled and the packet is longer than 2047 bytes then this field is 2047.)
Reserved	31:12	0x0	Reserved

8.20.5 Wakeup Packet Memory - WUPM (0x05A00 + 4*n [n=0...31]; RO)

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 is not cleared by any reset.

Field	Bit(s)	Initial Value	Description
WUPD	31:0	X	Wakeup Packet Data

8.20.6 IP Address Valid - IPAV (0x5838; R/W)

The IP Address Valid indicates whether the IP addresses in the IP Address Table are valid.

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.
Reserved	15:4	0x0	Reserved.
V60	16	0b	IPv6 Address 0 Valid.
Reserved	31:17	0b	Reserved.

8.20.7 IPv4 Address Table - IP4AT (0x05840 + 8*n [n=0...3]; R/W)

The IPv4 Address Table is used to store the four IPv4 addresses for the ARP/IPv4 Request packet and Directed IP packet wakeup.

Note: This table is not cleared by any reset.

Field	Bit(s)	Initial Value	Description
IP Address	31:0	X	IPv4 Address n

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV4ADDR0	0	0x5840	31:0	X	IPv4 Address 0
IPV4ADDR1	2	0x5848	31:0	X	IPv4 Address 1
IPV4ADDR2	4	0x5850	31:0	X	IPv4 Address 2
IPV4ADDR3	6	0x5858	31:0	X	IPv4 Address 3

8.20.8 IPv6 Address Table - IP6AT (0x05880 + 4*n [n=0...3]; R/W)

The IPv6 Address Table is used to store the IPv6 addresses for neighbor Discovery packet filtering and Directed IP packet wakeup.

Note: This table is not cleared by any reset.

Field	Bit(s)	Initial Value	Description
IP Address	31:0	X	IPv6 Address bytes 4*n+1:4*n +4

Field	Dword #	Address	Bit(s)	Initial Value	Description

IPV6ADDR0	0	0x5880	31:0	X	IPv6 Address 0, bytes 1-4
	1	0x5884	31:0	X	IPv6 Address 0, bytes 5-8
	2	0x5888	31:0	X	IPv6 Address 0, bytes 9-12
	3	0x588C	31:0	X	IPv6 Address 0, bytes 16-13

8.20.9 Flexible Host Filter Table Registers - FHFT (0x09000 - 0x093FC; RW)

Each of the 4 Flexible Host Filters Table registers (FHFT) contains a 128B pattern and a corresponding 128-bit mask array. If enabled, the first 128 bytes of the received packet are compared against the non-masked bytes in the FHFT register.

Each 128B filter is composed of 32 DW entries, where each 2 DWs are accompanied by an 8-bit mask, one bit per filter byte.

Note: The length field must be 8 bytes aligned. For filtering packets shorter than 8 bytes aligned the values should be rounded up to the next 8 bytes aligned value, the hardware implementation compares 8 bytes at a time so it should get extra zero masks (if needed) until the end of the length value.

The last DW of each filter contains a length field defining the number of bytes from the beginning of the packet compared by this filter, the length field should be 8 bytes aligned value. If actual packet length is less than (length - 8) (length is the value specified by the length field), the filter fails. Otherwise, it depends on the result of actual byte comparison. The value should not be greater than 128.

31 8	31 8	7 0	31 0	31 0
Reserved	Reserved	Mask [7:0]	DW 1	DW 0
Reserved	Reserved	Mask [15:8]	DW 3	DW 2
Reserved	Reserved	Mask [23:16]	DW 5	DW 4
Reserved	Reserved	Mask [31:24]	DW 7	DW 6

. . .

31 8	31 8	7 0	31 0	31 0
Reserved	Reserved	Mask [127:120]	DW 29	DW 28
Length	Reserved	Mask [127:120]	DW 31	DW 30

Field	Dword	Address	Bit(s)	Initial Value
Filter 0 DW0	0	0x09000	31:0	X
Filter 0 DW1	1	0x09004	31:0	X
Filter 0 Mask[7:0]	2	0x09008	7:0	X
Reserved	3	0x0900C		X
Filter 0 DW2	4	0x09010	31:0	X
...				

Filter 0 DW30	60	0x090F0	31:0	X
Filter 0 DW31	61	0x090F4	31:0	X
Filter 0 Mask[127:120]	62	0x090F8	7:0	X
Length	63	0x090FC	6:0	X

Accessing the FHFT registers during filter operation can result in a packet being mis-classified if the write operation collides with packet reception. It is therefore advised that the flex filters are disabled prior to changing their setup.

8.20.10 Flexible Host Filter Table Extended Registers - FHFT_EXT (0x09A00 - 0x09BFC; RW)

Each of the 2 additional Flexible Host Filters Table extended registers (FHFT_EXT) contains a 128B pattern and a corresponding 128-bit mask array. If enabled, the first 128 bytes of the received packet are compared against the non-masked bytes in the FHFT_EXT register. The layout and access rules of this table are the same as in FHFT.

8.21 Management Register Descriptions

All management registers are controlled by the remote BMC for both read and write. Host accesses to the management registers are blocked (read and write) unless debug write is enabled. The attributes for the fields in this chapter refer to the BMC access rights.

8.21.1 Management VLAN TAG Value - MAVTV (0x5010 +4*n [n=0...7]; RW)

Where “n” is the VLAN filter serial number, equal to 0,1,...7.

Field	Bit(s)	Initial Value	Description
VID	11:0	0x0	Contain the VLAN ID that should be compared with the incoming packet if the corresponding bit in MFVAL.VLAN is set.
Rsv	31:12	0x0	Reserved

The MAVTV registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets as described in the Management chapter.

8.21.2 Management Flex UDP/TCP Ports - MFUTP (0x5030 + 4*n [n=0...7]; RW)

Where each 32-bit register (n=0,...,7) refers to two port filters (register 0 refers to ports 0&1, register 2 refers to ports 2&3, etc.).

Field	Bit(s)	Initial Value	Description
MFUTP_even	15:0	0x0	i Management Flex UDP/TCP port
MFUTP_odd	31:16	0x0	i+1 Management Flex UDP/TCP port

The MFUTP registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See Section 10.4 .

Reset - The MFUTP registers are cleared on Internal_Power_On_Reset only. The initial values for this register can be loaded from the EEPROM after power-up reset.

Note: The MFUTP_even & MFUTP_odd fields should be written in network order.

8.21.3 Management Ethernet Type Filters- METF (0x5060 + 4*n [n=0...3]; RW)

Field	Bit(s)	Initial Value	Description
METF	15:0	0x0	EtherType value to be compared against the L2 EtherType field in the Rx packet.
Reserved	29:16	0x0	Reserved
Polarity	30	0b	
Reserved	31	0b	Reserved

The METF registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See Section 10.4 .

Reset - The METF registers are cleared on Internal_Power_On_Reset only. The initial values for this register might be loaded from the EEPROM after power-up reset.

8.21.4 Management Control Register - MANC (0x05820; RW)

Field	Bit(s)	Initial Value	Description
Reserved	15:0	0x0	Reserved.
TCO_RESET	16	0b	TCO Reset occurred. Set to 1b on a TCO Reset. This bit is only reset by Internal_Power_On_ResetInternal_Power_On_Reset.
RCV_TCO_EN	17	0b	Receive TCO Packets Enabled. When this bit is set it enables the receive flow from the wire to the MNG block.
keep_PHY_link_up	18	0b	Block PHY reset and power state changes. When this bit is set the PHY reset and power state changes does not get to the PHY, This bit can not be written unless Keep_PHY_Link_Up_En EEPROM bit is set. This bit is reset bit Internal_Power_On_Reset.
RCV_ALL	19	0b	Receive All Enable When set, all received packets that passed L2 filtering is directed to the MNG block.
RCV_ALL_MCST	20	0b	Receive all multicast: When set, all received multicast packets pass L2 filtering and might be directed to the MNG block by one of the decision filters. Broadcast packets are not forwarded by this bit.
EN_MNG2HOST	21	0b	Enable MNG packets to host memory This bit enables the functionality of the MANC2H register. When set the packets that are specified in the MANC2H registers is forwarded to the HOST memory too, if they pass manageability filters.
Bypass VLAN	22	0b	When set, VLAN filtering is bypassed for MNG packets.

EN_XSUM_FILTER	23	0b	Enable checksum filtering to MNG When this bit is set, only packets that pass L3,L4 checksum is sent to the MNG block.
EN_IPv4_FILTER	24	0b	Enable IPv4 address Filters – when set, the last 128 bits of the MIPAF register are used to store 4 IPv4 addresses for IPv4 filtering. When cleared, these bits store a single IPv6 filter.
FIXED_NET_TYPE	25	0b	Fixed net type: If set, only packets matching the net type defined by the NET_TYPE field passes to manageability. Otherwise, both tagged and un-tagged packets can be forwarded to the manageability engine.
NET_TYPE	26	0b	NET TYPE: 0b = pass only un-tagged packets. 1b = pass only VLAN tagged packets. Valid only if FIXED_NET_TYPE is set.
Reserved	27	0b	Reserved
Reserved	31:28	0b	Reserved

8.21.5 Manageability Filters Valid - MFVAL (0x5824; RW)

The Manageability filters valid register indicates which filter registers contain a valid entry.

Field	Bit(s)	Initial Value[1]	Description
MAC	3:0	0x0	MAC – Indicates if the MAC unicast filter registers (MMAH, MMAL) contain valid MAC addresses. Bit 0 corresponds to filter 0, etc.
Reserved	7:4	0x0	Reserved
VLAN	15:8	0x0	VLAN - Indicates if the VLAN filter registers (MAVTV) contain valid VLAN tags. Bit 8 corresponds to filter 0, etc.
IPv4	19:16	0x0	IPv4 - Indicates if the IPv4 address filters (MIPAF) contain valid IPv4 addresses. Bit 16 corresponds to IPv4 address 0. These bit apply only when IPv4 address filters are enabled. (MANC.EN_IPv4_FILTER=1b)
Reserved	23:20	0x0	Reserved
IPv6	27:24	0x0	IPv6 - Indicates if the IPv6 address filter registers (MIPAF) contain valid IPv6 addresses. Bit 24 corresponds to address 0, etc. Bit 27 (filter 3) applies only when IPv4 address filters are not enabled (MANC.EN_IPv4_FILTER=0b)
Reserved	31:28	0x0	Reserved

[1] The initial values for this register can be loaded from the EEPROM after power-up reset or firmware reset. The MFVAL register is written by the BMC and not accessible to the host for writing.

Reset - The MFVAL register is cleared on Internal_Power_On_Reset and firmware reset.

8.21.6 Management Control to Host Register - MANC2H (0x5860; RW)

The MANC2H register allows forwarding of manageability packets to the host based on the decision filter that forwarded it to the manageability micro-controller. Each manageability decision filter (MDEF) has a corresponding bit in the MANC2H register. When a manageability decision filter (MDEF) forwards a packet to manageability, it also forwards the packet to the host if the corresponding MANC2HOST bit is set and if the EN_MNG2HOST bit is set. The EN_MNG2HOST bit serves as a global enable for the MANC2H bits.

Field	Bit(s)	Initial Value[1]	Description
Host Enable	7:0	0x0	Host Enable – when set, indicates that packets forwarded by the manageability filters to manageability are also sent to the host. Bit 0 corresponds to decision rule 0, etc.
Reserved	31:8	0x0	Reserved

[1] Reset - The MANC2H register is cleared on Internal_Power_On_Reset and firmware reset. The initial values for this register can be loaded from the EEPROM after power-up reset or firmware reset.

8.21.7 Manageability Decision Filters- MDEF (0x5890 + 4*n [n=0...7]; RW)

Where "n" is the decision filter.

Field	Bit(s)	Initial Value[1]	Description
Unicast AND	0	0b	Unicast - Controls the inclusion of unicast address filtering in the manageability filter decision (AND section).
Broadcast AND	1	0b	Broadcast - Controls the inclusion of broadcast address filtering in the manageability filter decision (AND section).
VLAN AND	2	0b	VLAN - Controls the inclusion of VLAN address filtering in the manageability filter decision (AND section).
IP Address	3	0b	IP Address - Controls the inclusion of IP address filtering in the manageability filter decision (AND section).
Unicast OR	4	0b	Unicast - Controls the inclusion of unicast address filtering in the manageability filter decision (OR section).
Broadcast OR	5	0b	Broadcast - Controls the inclusion of broadcast address filtering in the manageability filter decision (OR section).
Multicast AND	6	0b	Multicast - Controls the inclusion of Multicast address filtering in the manageability filter decision (AND section). Broadcast packets are not included by this bit. The packet must pass some L2 filtering to be included by this bit – either by the MANC.MCST_PASS_L2 or by some dedicated MAC address.
ARP Request	7	0b	ARP Request - Controls the inclusion of ARP Request filtering in the manageability filter decision (OR section).
ARP Response	8	0b	ARP Response - Controls the inclusion of ARP Response filtering in the manageability filter decision (OR section).
neighbor Discovery	9	0b	neighbor Discovery - Controls the inclusion of neighbor Discovery filtering in the manageability filter decision (OR section). The neighbor types accepted by this filter are types 0x86, 0x87, 0x88 and 0x89.
Port 0x298	10	0b	Port 0x298 - Controls the inclusion of Port 0x298 filtering in the manageability filter decision (OR section).
Port 0x26F	11	0b	Port 0x26F - Controls the inclusion of Port 0x26F filtering in the manageability filter decision (OR section).
Flex port	27:12	0x0	Flex port - Controls the inclusion of Flex port filtering in the manageability filter decision (OR section). Bit 12 corresponds to flex port 0, etc.
Flex TCO	31:28	0x0	Flex TCO - Controls the inclusion of Flex TCO filtering in the manageability filter decision (OR section). Bit 28 corresponds to Flex TCO filter 0, etc.

[1] Default values are read from EEPROM.

8.21.8 Manageability Decision Filters- MDEF_EXT (0x5930 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value[1]	Description
L2 EtherType AND	3:0	0x0	L2 EtherType - Controls the inclusion of L2 EtherType filtering in the manageability filter decision (AND section).
Reserved	7:4	0x0	Reserved for additional L2 EtherType AND filters.
L2 EtherType OR	11:8	0x0	L2 EtherType - Controls the inclusion of L2 EtherType filtering in the manageability filter decision (OR section).
Reserved	15:12	0x0	Reserved for additional L2 EtherType OR filters.
Reserved	31:16	0x0	Reserved

[1] Default values are read from EEPROM.

8.21.9 Manageability IP Address Filter - MIPAF (0x58B0 + 4*n [n=0...15]; RW)

The Manageability IP Address Filter register stores IP addresses for manageability filtering. The MIPAF register can be used in two configurations, depending on the value of the MANC. EN_IPv4_FILTER bit:

- EN_IPv4_FILTER = 0: the last 128 bits of the register store a single IPv6 address (IPV6ADDR3)
- EN_IPv4_FILTER = 1: the last 128 bits of the register store 4 IPv4 addresses (IPV4ADDR[3:0])

EN_IPv4_FILTER = 0:

DWORD#	Address	31	0
0	0x58B0	IPV6ADDR0	
1	0x58B4		
2	0x58B8		
3	0x58BC		
4	0x58C0	IPV6ADDR1	
5	0x58C4		
6	0x58C8		
7	0x58CC		
8	0x58D0	IPV6ADDR2	
9	0x58D4		
10	0x58D8		
11	0x58DC		
12	0x58E0	IPV6ADDR3	
13	0x58E4		
14	0x58E8		
15	0x58EC		

Field definitions for 0 setting:

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV6ADDR0	0	0x58B0	31:0	X*	IPv6 Address 0, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58B4	31:0	X*	IPv6 Address 0, bytes 5-8
	2	0x58B8	31:0	X*	IPv6 Address 0, bytes 9-12
	3	0x58BC	31:0	X*	IPv6 Address 0, bytes 16-13
IPV6ADDR1	0	0x58C0	31:0	X*	IPv6 Address 1, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58C4	31:0	X*	IPv6 Address 1, bytes 5-8
	2	0x58C8	31:0	X*	IPv6 Address 1, bytes 9-12
	3	0x58CC	31:0	X*	IPv6 Address 1, bytes 16-13
IPV6ADDR2	0	0x58D0	31:0	X*	IPv6 Address 2, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58D4	31:0	X*	IPv6 Address 2, bytes 5-8
	2	0x58D8	31:0	X*	IPv6 Address 2, bytes 9-12
	3	0x58DC	31:0	X*	IPv6 Address 2, bytes 16-13
IPV6ADDR3	0	0x58E0	31:0	X*	IPv6 Address 3, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58E4	31:0	X*	IPv6 Address 3, bytes 5-8
	2	0x58E8	31:0	X*	IPv6 Address 3, bytes 9-12
	3	0x58EC	31:0	X*	IPv6 Address 3, bytes 16-13

EN_IPv4_FILTER = 1:

DWORD#	Address	31	0
0	0x58B0	IPV6ADDR0	
1	0x58B4		
2	0x58B8		
3	0x58BC		
4	0x58C0	IPV6ADDR1	
5	0x58C4		
6	0x58C8		
7	0x58CC		
8	0x58D0	IPV6ADDR2	
9	0x58D4		
10	0x58D8		
11	0x58DC		
12	0x58E0	IPV4ADDR0	
13	0x58E4	IPV4ADDR1	
14	0x58E8	IPV4ADDR2	
15	0x58EC	IPV4ADDR3	

Field definitions for 1 Setting:

Field	Dword #	Address	Bit(s)	Initial Value[1]	Description
IPV6ADDR0	0	0x58B0	31:0	X	IPv6 Address 0, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58B4	31:0	X	IPv6 Address 0, bytes 5-8
	2	0x58B8	31:0	X	IPv6 Address 0, bytes 9-12
	3	0x58BC	31:0	X	IPv6 Address 0, bytes 16-13
IPV6ADDR1	0	0x58C0	31:0	X	IPv6 Address 1, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58C4	31:0	X	IPv6 Address 1, bytes 5-8
	2	0x58C8	31:0	X	IPv6 Address 1, bytes 9-12
	3	0x58CC	31:0	X	IPv6 Address 1, bytes 16-13
IPV6ADDR2	0	0x58D0	31:0	X	IPv6 Address 2, bytes 1-4 (L.S. byte is first on the wire)
	1	0x58D4	31:0	X	IPv6 Address 2, bytes 5-8
	2	0x58D8	31:0	X	IPv6 Address 2, bytes 9-12
	3	0x58DC	31:0	X	IPv6 Address 2, bytes 16-13
IPV4ADDR0	0	0x58E0	31:0	X	IPv4 Address 0 (L.S. byte is first on the wire)
IPV4ADDR1	1	0x58E4	31:0	X	IPv4 Address 1 (L.S. byte is first on the wire)
IPV4ADDR2	2	0x58E8	31:0	X	IPv4 Address 2 (L.S. byte is first on the wire)
IPV4ADDR3	3	0x58EC	31:0	X	IPv4 Address 3 (L.S. byte is first on the wire)

[1] The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

Initial value:

Field	Bit(s)	Initial Value[1]	Description
IP_ADDR 4 bytes	31:0	X	4 bytes of IP (v6 or v4) address. i mod 4 = 0 to bytes 1 – 4 i mod 4 = 1 to bytes 5 – 8 i mod 4 = 0 to bytes 9 – 12 i mod 4 = 0 to bytes 13 – 16 where i div 4 is the index of IP address (0...3)

[1] The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

Reset - The registers are cleared on Internal_Power_On_Reset only.

Note: These registers should be written in network order.

8.21.10 Manageability MAC Address Low - MMAL (0x5910 + 8*n [n= 0...3]; RW)

Where “n” is the exact unicast/Multicast address entry, equal to 0,1,...3.

Field	Bit(s)	Initial Value[1]	Description
MMAL	31:0	X	Manageability MAC Address Low. The lower 32 bits of the 48 bit Ethernet address.

[1] The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

These registers contain the lower bits of the 48 bit Ethernet address. The MMAL registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See Section 10.4 .

Reset - The MMAL registers are cleared on Internal_Power_On_Reset only. The initial values for this register can be loaded from the EEPROM after power-up reset.

Note: The MMAL.MMAL field should be written in network order.

8.21.11 Manageability MAC Address High - MMAH (0x5914 + 8*n [n=0...3]; RW)

Where “n” is the exact unicast/Multicast address entry, equal to 0,1,...3.

Field	Bit(s)	Initial Value[1]	Description
MMAH	15:0	X	**Manageability MAC Address High**. The upper 16 bits of the 48 bit Ethernet address.
Reserved	31:16	0x0	**Reserved**. Reads as 0. Ignored on write.

[1] The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

These registers contain the upper bits of the 48 bit Ethernet address. The complete address is {MMAH, MMAL}. The MMAH registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See Section 10.4 .

Reset - The MMAL registers are cleared on Internal_Power_On_Reset only. The initial values for this register can be loaded from the EEPROM after power-up reset or firmware reset.

Note: The MMAH.MMAH field should be written in network order.

8.21.12 Flexible TCO Filter Table registers - FTFT (0x09400-0x097FC; RW)

Each of the 4 Flexible TCO Filters Table registers (FTFT) contains a 128 byte pattern and a corresponding 128-bit mask array. If enabled, the first 128 bytes of the received packet are compared against the non-masked bytes in the FTFT register.

Each 128B filter is composed of 32 DW entries, where each 2 DWs are accompanied by an 8-bit mask, one bit per filter byte. The bytes in each 2 DWs are written in network order i.e. byte0 written to bits [7:0], byte1 to bits [15:8] etc. The mask field is set so that bit 0 in the mask masks byte 0, bit 1 masks byte 1 etc. A value of 1 in the mask field means that the appropriate byte in the filter should be compared to the appropriate byte in the incoming packet.

The last DW of each filter contains a length field defining the number of bytes from the beginning of the packet compared by this filter. If the actual packet length is less than the length specified by this field, the filter fails. Otherwise, it depends on the result of the actual byte comparison. The length field should not be greater than 128 and not smaller than 8.

Note: The mask field and length field should be 8 bytes aligned.

The packet length examined by the filter includes the 4 bytes of the CRC, even if the CRC is stripped.

31 0	31 8	7 0	31 0	31 0
Reserved	Reserved	Mask [7:0]	DW 1	DW 0
Reserved	Reserved	Mask [15:8]	DW 3	DW 2
Reserved	Reserved	Mask [23:16]	DW 5	DW 4
Reserved	Reserved	Mask [31:24]	DW 7	DW 6

. . .

31 0	31 8	7 0	31 0	31 0
Reserved	Reserved	Mask [127:120]	DW 29	DW 28
Length	Reserved	Mask [127:120]	DW 31	DW 30

Field definitions for Filter Table Registers:

Field	Dword	Address	Bit(s)	Initial Value
Filter 0 DW0	0	0x09400	31:0	X
Filter 0 DW1	1	0x09404	31:0	X
Filter 0 Mask[7:0]	2	0x09408	7:0	X
Reserved	3	0x0940C		X
Filter 0 DW2	4	0x09410	31:0	X
...				
Filter 0 DW30	60	0x094F0	31:0	X
Filter 0 DW31	61	0x094F4	31:0	X
Filter 0 Mask[127:120]	62	0x094F8	7:0	X
Length	63	0x094FC	6:0	X

The initial values for the FTFT registers can be loaded from the EEPROM after power-up reset. The FTFT registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets as described in Section 10.4.3.5.

Reset - The FTFT registers are cleared on Internal_Power_On_Reset only.

8.22 IPsec Registers Description

IPsec registers are owned by the PF in an IOV mode.

There is no added value here to encrypt the SA contents when being read by software, because the SA content is available in clear text from the system memory, like any IPsec flow handled in software.

8.22.1 IPSec Control – IPSCTRL (0xB430; RW)

Field	Bit(s)	Initial Value	Description
TX_IPSEC_EN	0	0b RW	IPsec Tx offload Enable bit. When set, IPsec offload ability is enabled for Tx path. When cleared, IPsec offload ability is disabled for Tx path, regardless of the contents of the Tx SA table.
RX_IPSEC_EN	1	0b RW	IPsec Rx offload Enable bit. When set, IPsec offload ability is enabled for Rx path. When cleared, IPsec offload ability is disabled for Rx path, regardless of the contents of Rx SA tables.
DELETE_ALL	2	0b RW/SC	Delete All bit. When set, the hardware invalidates all the Rx table entries. This bit can be set by SW only if IPSRXCMD.BUSY bit was read as cleared before. The Delete All bit is cleared when hardware ends deleting all the entries.
IPSEC_FRAG	3	0b	Avoid IPsec offload on IP fragments. 0 = no IPsec offload is done on IP fragments. Correct operating mode. 1 = IPsec offload is done even on IP fragments.
Reserved	31:4	0x0	Reserved

8.22.2 IPsec Tx Index - IPSTXIDX (0xB450; RW)

Field	Bit(s)	Initial Value	Description
SA_IDX	7:0	0x0	SA index for indirect access into the Tx SA table.
Reserved	31:8	0x0	Reserved

8.22.3 IPsec Tx Key Registers - IPSTXKEY (0xB460 + 4*n [n = 0...3]; RW)

Defines the KEY value used as part of the Tx SA. See Section 7.8.1.5.1 for details

Field	Bit(s)	Initial Value	Description

AES-128 KEY	31:0	0x0	4 bytes of 16 bytes key that has been read/written from/into the Tx SA entry pointed by SA_IDX.
			n=0 contains the LSB of the key.
			n=3 contains the MSB of the key.
			Any write in this register must be followed by a write in the IPSTXSALT register, as it is the trigger for internal write of the whole entry into the Tx SA Table.

8.22.4 IPsec Tx Salt Register - IPSTXSALT (0xB454; RW)

Defines the SALT value used as part of the Tx SA. See Section 7.8.1.5.1 for details.

Field	Bit(s)	Initial Value	Description
AES-128 SALT	31:0	0x0	4 bytes salt that has been read/written from/into the Tx SA entry pointed by SA_IDX. Writing this register is used internally to trigger the write of the whole entry into the Tx SA Table, and should thus be written last whenever updating a Tx SA entry.

8.22.5 IPsec Rx Command Register - IPSRXCMD (0xB408; RW)

Field	Bit(s)	Initial Value	Description
Reserved	1:0	00b	Reserved
PROTO	2	0b	IPsec Protocol select. When set this SA offloads ESP packets. When reset this SA offloads AH packets.
DECRYPT	3	0b	When set hardware performs decrypting offload on this SA. Meaningful only if PROTO is set (ESP mode).
IPv6	4	0b	IPv6 type. When set this SA expects to receive an IPv6 packet. When cleared, hardware forces the IPv6 address bit IPSRXIPADDR 0..2 to 0 and the look up is based only on the high order 32 bits (IPSRXIPADDR3). This bit is the LS-bit in the search key. When software adds an entry it must write all the IPSRXIPADDR 0...3 in any case (IPv6 or IPv4). In case of an IPv4 address, IPSRXIPADDR 0..2 must be set to 0.
Reserved	7:5	000b	Reserved
USED_SA (RO)	16:8	0x0	Number of used SA 0...256. If equal to 0 delete commands is ignored. If equal to 256 add commands is ignored.
Reserved	29:17	0x0	Reserved
ADD_DEL	30	0b	Add or delete command. When set hardware adds the SA information to the Rx SA table. When clear the hardware deletes the SA that matches the IPSRXSPI, the IPSRXADDR 0...3 registers, and the IPv6 type bit.
BUSY (RO/SC)	31	0b	Busy bit. Used by hardware to lock software access to the SA table while the hardware is in adding or deleting process. The BUSY bit is automatically set by hardware when software writes to this register. The BUSY bit is cleared when hardware ends adding or deleting an SA.

Notes: If software reads the BUSY bit as set, any IPsec register write has no effect, and software should continue polling the IPSRXCMD.BUSY bit (read access) until it is read as cleared.

Software should not make changes in the Tx SA Table while changing the IPSEC_EN bit.

8.22.6 IPsec Rx SPI Register - IPSRXSPI (0xB40C; RW)

Field	Bit(s)	Initial Value	Description
SPI	31:0	0x0	SPI field that has been deleted/added from/into the Rx SA table. **Note**: Field is defined in Big Endian (MS byte is first on the wire)

8.22.7 IPsec Rx Key Register - IPSRXKEY (0xB410 + 4 * n [n = 0..3]; RW)

Field	Bit(s)	Initial Value	Description

AES-128 KEY	31:0	0x0	4 bytes of 16 bytes key that has been deleted/added from/into the Rx SA table.
			n=0 contains the LSB of the key.
			n=3 contains the MSB of the key.

8.22.8 IPsec Rx Salt Register - IPSRXSALT (0xB404; RW)

Field	Bit(s)	Initial Value	Description
AES-128 SALT	31:0	0x0	4 bytes salt that has been deleted/added from/into the Rx SA table.

8.22.9 IPsec Rx IP address Register - IPSRXIPADDR (0xB420 + 4*n [n = 0..3]; RW)

Field	Bit(s)	Initial Value	Description
IPADDR	31:0	0x0	4 bytes of 16 bytes destination IP address for the associated Rx SA(s) that has been deleted/added from/into the Rx SA table. n=0 contains the MSB of an IPv6 address. n=3 contains the LSB of an IPv6 address or an IPv4 address. For an IPv4 address, IPSRXIPADDR 0...2 shall be written by SW with zeros. **Note**: Field is defined in Big Endian (LS byte is first on the wire).

8.22.10 IPsec Rx Index - IPSRXIDX (0xB400; RW)

Field	Bit(s)	Initial Value	Description
SA_IDX	7:0	0x0	SA index for indirect access into the Rx SA table.
Reserved	30:8	0x0	Reserved
DBG_MOD	31	0b	Debugging read access mode. When set, the Rx SA entries can be read via setting the SA_IDX field to the SA entry index to be read. When cleared, the normal Rx SA table access mode is used. This bit must be set only while the BUSY bit of the IPSRXCMD register was read as cleared.

8.23 Diagnostic Registers Description

The 82576 contains several diagnostic registers. These registers enable software to directly access the contents of the 82576's internal Packet Buffer Memory (PBM), also referred to as FIFO space. These registers also give software visibility into what locations in the PBM that the hardware currently considers to be the "head" and "tail" for both transmit and receive operations.

8.23.1 Receive Data FIFO Head Register - RDFH (0x02410; RWS)

This register stores the head of the on–chip receive data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Receive FIFO Head. So a value of "0x8" in this register corresponds to an offset of 8 quadwords into the Receive FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO 0 Head	12:0	0b	Receive FIFO 0 Head pointer. This field refers to the whole RX packet buffer. Note: The field is in units of 64-bit lines.
Reserved	30:13	0b	Reads as 0b. Should be written to 0b for future compatibility.
FIFO 0 Full	31	0b	Rx Memory Full Signal - This bit rises when there are less than 4 empty rows in the RX packet buffer. This bit indicates the status of the whole RX packet buffer.

8.23.2 Receive Data FIFO Tail Register - RDFT (0x02418; RWS)

This register stores the tail of the on–chip receive data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Receive FIFO Tail. So a value of "0x8" in this register corresponds to an offset of eight quadwords or into the Receive FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Tail 0	12:0	0b	Receive FIFO buffer Tail pointer. This field refers to the whole RX packet buffer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

8.23.3 Receive Data FIFO Head Saved Register - RDFHS (0x2420; RWS)

This register stores a copy of the Receive Data FIFO Head register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Head 0	12:0	0b	A "saved" value of the Receive FIFO Head pointer . This field refers to the whole RX packet buffer. Note: The field is in units of 64-bit lines.
Reserved	31:15	0b	Reads as 0b. Should be written to 0b for future compatibility.

8.23.4 Receive Data FIFO Tail Saved Register - RDFTS (0x02428; RWS)

This register stores a copy of the Receive Data FIFO Tail register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Tail 0	12:0	0b	RX Read desc pointer. This field refers to the whole RX packet buffer. Note: The field is in units of 64-bit lines.
Reserved	31:15	0b	Reads as 0b. Should be written to 0b for future compatibility.

8.23.5 Switch Buffer FIFO Head Register - SWBFH (0x03010; RWS)

Field	Bit(s)	Initial Value	Description
FIFO 0 Head	12:0	0b	switch FIFO 0 Head pointer. This field refers to the whole switch packet buffer. Note: The field is in units of 64-bit lines.
Reserved	30:13	0b	Should be written to 0b for future compatibility.
FIFO 0 Full	31	0b	Switch Memory Full Signal - This bit rises when there are less than 4 empty rows in the RX packet buffer. This bit indicates the status of the whole switch packet buffer.

This register stores the head pointer of the on-chip switch data FIFO. Since the internal FIFO is organized in units of 64 bit words, this field contains the 64 bit offset of the current switch FIFO Head. So a value of “0x8” in this register corresponds to an offset of 8 QWORDS or 64 bytes into the switch FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

8.23.6 Switch Buffer FIFO Tail Register - SWBFT (0x03018; RWS)

Field	Bit(s)	Initial Value	Description
FIFO Tail 0	12:0	0b	Switch FIFO Tail pointer. This field refers to the whole switch packet buffer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

This register stores the tail pointer of the on-chip switch data FIFO. Since the internal FIFO is organized in units of 64 bit words, this field contains the 64 bit offset of the current switch FIFO Tail. So a value of “0x8” in this register corresponds to an offset of 8 QWORDS or 64 bytes into the switch FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

8.23.7 Switch Buffers FIFO Head Saved Register - SWBFHS (0x03020; RWS)

Field	Bit(s)	Initial Value	Description
FIFO Head 0	12:0	0b	A “saved” value of the Switch FIFO Head pointer. This field refers to the whole switch packet buffer. Note: The field is in units of 64-bit lines.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

8.23.8 Switch Buffers FIFO Tail Saved Register - SWBFTS (0x03028; RWS)

Field	Bit(s)	Initial Value	Description
FIFO Tail 0	12:0	0b	Switch Read desc pointer0. This field refers to the whole Switch packet buffer. Note: The field is in units of 64-bit lines.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

8.23.9 Packet Buffer Diagnostic - PBDIAG (0x02458; R/W)

Field	Bit(s)	Initial Value	Description
Bypass mode	0	0b	Descriptor Monitor Bypass. When this bit is set to 1b the descriptor monitor (checking if there is enough descriptors in Q ring) is disabled.
PB_MNG	1	0b	**Packet Buffer for Manageability:** when set to 1b all Rx traffic is not written to the packet buffer so that the packet buffer could be used as memory for manageability controller code.
Reserved	19:2	0x0	Reserved.
DBU_empty (RO)	20	X	All FIFOs (Rx and Tx) are empty.
Cfg_rx_wait	21	0b	Stop reading data from the receive data buffer to the DMA Rx machine. Diagnostic only.
Cfg_tx_wait	22	0b	Stop reading data from the transmit data buffer towards the Tx MAC. Diagnostic only
Reserved	28:23	00b	Reserved Always set to 00b.
STAT_SEL	31:29	0x0	Select the statistics reflected in DBGC_1 to DBGC_3

8.23.10 Transmit Data FIFO Head Register - TDFH (0x03410; RWS)

This register stores the head of the on–chip transmit data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Transmit FIFO Head. A value of 0x8 in this register corresponds to an offset of 8 quadwords into the Transmit FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Head 0	12:0	0b	Transmit FIFO Head pointer. This field refers to the whole TX packet buffer. Note: The field is in units of 64-bit lines.
Reserved	30:13	0x0	Reads as 0b. Should be written to 0b for future compatibility.
Tx Memory Full 0	31	0b	Tx FIFO memory full indication. This bit rises when there are less than 4 empty rows in the TX packet buffer. This bit indicates the status of the whole TX packet buffer.

8.23.11 Transmit Data FIFO Tail Register - TDFT (0x03418; RWS)

This register stores the tail of the on–chip transmit data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Transmit FIFO Tail. A value of 0x8 in this register corresponds to an offset of 8 quadwords into the Transmit FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Tail 0	12:0	0x0	Transmit FIFO tail pointer. This field refers to the whole TX packet buffer. Note: The field is in units of 64-bit lines.
Reserved	30:13	0x0	Reads as 0b. Should be written to 0b for future compatibility.

8.23.12 Transmit Data FIFO Head Saved Register - TDFHS (0x03420; RWS)

This register stores a copy of the Transmit Data FIFO Head register in case the internal register needs to be restored. This register points to the beginning of the last packet in the packet buffer, even if it was already transmitted. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Head 0	12:0	0x0	Transmit FIFO Last Packet Header Pointer. This field refers to the whole TX packet buffer. Note: The field is in units of 64-bit lines
Reserved	31:13	000b	Reads as 000b. Should be written to 0b for future compatibility.

8.23.13 Transmit Data FIFO Tail Saved Register - TDFTS (0x03428; RWS)

This register stores a copy of the Transmit Data FIFO Tail register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
FIFO Tail 0	12:0	0x0	TX Read desc pointer. This field refers to the whole RX packet buffer. Note: The field is in units of 64-bit lines.
Reserved	31:13	000b	Reads as 000b. Should be written to 0b for future compatibility.

8.23.14 Transmit Data FIFO Packet Count - TDFPC (0x03430; RO)

This register reflects the number of packets to be transmitted that are currently in the Transmit FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
TX FIFO Packet Count 0	11:0	0x0	The number of packets to be transmitted that are currently in the TX FIFO - Reflects the number of packets in the full packet buffer.
Reserved	31:13	0000b	Reads as 0000b.

8.23.15 Receive Data FIFO Packet Count - RDFPC (0x02430; RO)

This register reflects the number of packets to be received that are currently in the Receive FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
RX FIFO Packet Count 0	11:0	0x0	The number of packets to be received that are currently in theRX FIFO - Reflects the number of packets in the full packet buffer.
Reserved	31:12	0000b	Reads as 0000b.

8.23.16 Switch Data FIFO Packet Count - SWDFPC (0x03030; RO)

This register reflects the number of packets to be received that are currently in the switch FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.

Field	Bit(s)	Initial Value	Description
Software FIFO Packet Count 0	11:0	0x0	The number of packets to be received that are currently in theswitch FIFO - Reflects the number of packets in the full packet buffer.
Reserved	31:12	0000b	Reads as 0000b.

8.23.17 IpSec Packet Buffer ECC Status - IPPBECCSTS (0xB470; RC)

8.23.18 PB Slave Access Control - PBSLAC (0x3100; RW)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads.
Uncorr_err_cnt	15:8	0x0	Uncorrectable Error Count This counter is incremented every time an uncorrectable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads.
ECC Enable (RW)	16	1b	ECC Enable for Packet Buffer.
Reserved	25:17	0b	Reserved
Pb_cor_err_sta	26	0b	Status of PB Correctable Error. This bit is cleared by a read.
Pb_uncor_err_sta	27	0b	Status of PB Uncorrectable Error. This bit is cleared by a read.
Reserved	31:28	0b	Reserved

All PBM (FIFO) data is available to diagnostics. Locations are accessed as 128-bit words using the PBSLAC & PBSLAD registers.

Field	Bit(s)	Initial Value	Description
Reserved	3:0	0x0	Reserved. Must be written with 0.
Addr	15:4	0x0	Address. Software sets the address in which to access the chosen memory. Aligned for 16 bytes.
Reserved	18:16	0x0	Reserved.
Mem_Sel	20:19	0x0	Memory Select. 00 – RX PB 01 – TX PB 10 – Switch PB 11 - Reserved
Reserved	29:21	0x0	Reserved
Rd_Req	30	0x0	Read Request. The software sets this bit when asking to read data. hardware clears this bit when data is ready. Reading can be done during traffic.
Wr_Req	31	0x0	Write Request. The software sets this bit when asking to write data. hardware clears this bit when data is ready. Writing should not be done during traffic.

8.23.19 PB Slave Access Data – PBSLAD (0x3110 + 4*n [n= 0...3]; RW)

Field	Bit(s)	Initial Value	Description

Data	31:0	0x0	Data. Packet buffer RD/WR data
			n = 0: bits [31:0]
			n = 1: bits [63:32]
			n = 2: bits [95:64]
			n = 3: bits [127:96]

8.23.20 Rx Descriptor Handler Memory - RDHM (0x06000 + 4*n [n= 0..1023]; RO)

Field	Bit(s)	Initial Value	Description
FIFO Data	31:0	X	Rx Descriptor Handler Data

All the Rx descriptor handler cache is available to diagnostics. Locations can be accessed as 32 bit or 64 bit words. The descriptor handler cache is 4KB in size. The Cache is divided to 4 queues of 1K each. The access to the memory is linear. For example, descriptor N (31:0) in Rx queue X (15:0) is located at address 0x06000 + X * 0x200 + N * 0x10.

The packet buffer is accessible by pages of 4K. The accessed page is set in the RDHMP register.

8.23.21 Rx Descriptor Handler Memory Page Number - RDHMP (0x025FC; RW)

Field	Bit(s)	Initial Value	Description
Page	3:0	0x0	Rx Descriptor Handler Accessed Page (4KB) Valid values are 0 or 1.
Reserved	29:4	0x0	Reserved
Queue Depth	31:30	00b	Defines the number of descriptors (per queue) in the cache. 00b = 32 descriptors 01b = 16 descriptors 10b = 8 descriptors 11b = 4 descriptors

Note: The queue depth field must be updated before the receive queues are enabled (before writing to any CSR that controls the queues).

8.23.22 Tx Descriptor Handler Memory - TDHM (0x07000 + 4*n [n= 0..1023]; RO)

Field	Bit(s)	Initial Value	Description
FIFO Data	31:0	X	Rx Descriptor Handler Data

All the Tx descriptor handler cache is available to diagnostics. Locations can be accessed as 32 bit or 64 bit words. The descriptor handler cache is 4KB in size. The Cache is divided to 4 queues of 1K each. The access to the memory is linear. For example, descriptor N (31:0) in Tx queue X (15:0) is located at address 0x07000 + X * 0x200 + N * 0x10.

The packet buffer is accessible by pages of 4K. The accessed page is set in the RDHMP register.

8.23.23 Tx Descriptor Handler Memory Page Number - TDHMP (0x035FC; R/W)

Field	Bit(s)	Initial Value	Description
Page	3:0	0x0	Tx Descriptor Handler Accessed Page (4KB) Valid values are 0 or 1.
Reserved	29:4	0x0	Reserved
Queue Depth	31:30	00b	Defines the number of descriptors (per queue) in the cache. 00b = 32 descriptors 01b = 16 descriptors 10b = 8 descriptors 11b = 4 descriptors

Note: The queue depth field must be updated before the receive queues are enabled (before writing to any CSR that controls the queues).

The PBTCWBCOL, PBTCCOMP, PBTCWBD, PBTCWBRS, PBTCWBTAIL bits enable performance of internal descriptor handler operations in parallel to the software tail bumping process. Thus avoiding a performance impact when the tail is bumped frequently.

8.23.24 Rx Packet Buffer ECC Status - RPBECCSTS (0x0245C; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads.
Uncorr_err_cnt	15:8	0x0	Uncorrectable Error Count This counter is incremented every time an uncorrectable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads. Note that this counter might count the same error more than once if the data is read multiple time.
ECC Enable (RW)	16	1b	ECC Enable for Packet Buffer.
Reserved	25:17	0x0	Reserved
Pb_cor_err_sta	26	0b	Status of PB Correctable Error. This bit is cleared by a read.
Pb_uncor_err_sta	27	0b	Status of PB Uncorrectable Error. This bit is cleared by a read.
Reserved	31:28	0x0	Reserved

Note: Header replication, header split, or packets replication may cause the respective counters to count more than once.

8.23.25 Tx Packet Buffer ECC Status - TPBECCSTS (0x0345C; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads.
Uncorr_err_cnt	15:8	0x0	Uncorrectable Error Count This counter is incremented every time an uncorrectable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads.
ECC Enable (RW)	16	1b	ECC Enable for Packet Buffer.
Reserved	25:17	0x0	Reserved
Pb_cor_err_sta	26	0b	Status of PB Correctable Error. This bit is cleared by a read.
Pb_uncor_err_sta	27	0b	Status of PB Uncorrectable Error. This bit is cleared by a read.
Reserved	31:28	0x0	Reserved

Note: Uncorrectable errors occuring in the Tx packet buffer may propagate to the Switch buffer.

8.23.26 Switch Packet Buffer ECC Status - SWPBECCSTS (0x0305C; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads.
Uncorr_err_cnt	15:8	0x0	Uncorrectable Error Count This counter is incremented every time an uncorrectable error is detected; the counter stops after reaching 0xFF. These bits are cleared by reads. Note that this counter might count errors propagated from the Tx buffer and might count the same error more than once if the data is read multiple time.
ECC Enable (RW)	16	1b	ECC Enable for Packet Buffer.
Reserved	25:17	0x0	Reserved
Pb_cor_err_sta	26	0b	Status of PB Correctable Error. This bit is cleared by a read.
Pb_uncor_err_sta	27	0b	Status of PB Uncorrectable Error. This bit is cleared by a read.
Reserved	31:28	0x0	Reserved

8.23.27 IPSec Packet Buffer ECC Error Inject - IPPBEEI (0xB474; RW)

Field	Bit(s)	Initial Value	Description
Write Injection Enable	0	0b	Write Injection Enable When this bit is set, an error is injected the next time a line is written to the packet buffer. This bit is auto cleared by hardware when an error is injected.
Reset Data	1	0b	Reset Data Clears all bits in the data line on which the error is inserted.
Reserved	15:2	0x0	Reserved
Error1 Bit Location	23:16	0xFF	No Error Injection on This Bit Maximum allowed value is 74 for error injection.
Error2 Bit Location	31:24	0xFF	No Error Injection on This Bit Maximum allowed value is 74 for error injection.

8.23.28 Rx Descriptor Handler ECC Status - RDHESTS (0x025C0; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected in the Rx descriptor handler memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
Uncorr_err_cnt	15:8	0x0	Uncorrectable Error Count This counter is incremented every time an uncorrectable error is detected in the Rx descriptor handler memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
RDHECC Enable (RW)	16	1b	Rx Descriptor Handler ECC Enable
Reserved	31:17	0x0	Reserved

Note: A single error may cause the relevant counter to increase more than once.

8.23.29 Tx Descriptor Handler ECC Status - TDHESTS (0x35C0; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected in the Tx descriptor handler memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
Uncorr_err_cnt	15:8	0x0	Uncorrectable Error Count This counter is incremented every time an uncorrectable error is detected in the Tx descriptor handler memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
TDHECC Enable (RW)	16	1b	Tx Descriptor Handler ECC Enable
Reserved	31:17	0x0	Reserved

Note: A single error may cause the relevant counter to increase more than once.

8.23.30 PCIe Retry Buffer ECC Status - PRBESTS (0x05BA0; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected in the PCIe retry buffer memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
Reserved	15:8	0x0	Reserved
PRBECC Enable (RW)	16	1b	PCIe Retry Buffer ECC Enable
Reserved	31:17	0x0	Reserved

8.23.31 PCIe Write Buffer ECC Status - PWBESTS (0x05BB0; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected in the PCIe write buffer memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
Reserved	15:8	0x0	Reserved
PWBECC Enable (RW)	16	1b	PCIe Write Buffer ECC Enable
Reserved	31:17	0x0	Reserved

8.23.32 PCIe MSI-X ECC Status - PMSIXESTS (0x05BA8; RC)

Field	Bit(s)	Initial Value	Description
Corr_err_cnt	7:0	0x0	Correctable Error Count This counter is incremented every time a correctable error is detected in the PCIe MSI-X vector memory; the counter stops after reaching 0xFF. These bits are cleared by reads.
Reserved	15:8	0x0	Reserved
PMSIXE Enable (RW)	16	1b	PCIe MSI-X memory ECC Enable
Reserved	31:17	0x0	Reserved

8.23.33 Parity and ECC Error Indication- PEIND (0x1084; RC)

Field	Bit(s)	Initial Value	Description
dbu_ecc_sw_pb_nfatal	0	0b	Non fatal error detected in the data part of a packet in the switch packet buffer memory
dbu_ecc_tx_pb_nfatal	1	0b	Non fatal error detected in the data part of a packet in the Tx packet buffer memory
dbu_ecc_rx_pb_nfatal	2	0b	Non fatal error detected in the data part of a packet in the Rx packet buffer memory
Reserved	7:3	0x0	Reserved
dtx_parity_temp_fatal	8	0b	Fatal error detected in TSO prototype header memory
mac_parity_secu_rx_sa_fatal	9	0b	Fatal error detected in IPSec Rx keys memory
mac_parity_secu_tx_sa_fatal	10	0b	Fatal error detected in IPSec Tx keys memory
drx_parity_desc_fifo_fatal	11	0b	Fatal error detected in internal received packets descriptor memory
dtx_parity_dpt2_fatal	12	0b	Fatal error detected in internal transmit packets descriptor memory
dtx_parity_hdr_fatal	13	0b	Fatal error detected in internal transmit packets descriptor memory
dbu_parity_drx_pb_reg_fatal	14	0b	Fatal error detected in Rx packet buffer registers file
dtx_parity_dh_reg_fatal	15	0b	Fatal error detected in Tx descriptor handler registers file
drx_parity_dh_reg_fatal	16	0b	Fatal error detected in Rx descriptor handler registers file
dtx_parity_lso_fatal	17	0b	Fatal error detected in TSO internal Tx context memory
dtx_parity_cntxt_fatal	18	0b	Fatal error detected in internal Tx context memory
xtx_parity_buf_stg_fatal	19	0b	Fatal error detected in XTX internal FIFO
mac_parity_secu_post_l4cs_fatal	20	0b	Fatal error detected in IPSec Checksum Rx FIFO
mac_parity_secu_pre_hdr_parse_fatal	21	0b	Fatal error detected in IPSec pre-processing Rx FIFO
mac_parity_secu_post_sta_fatal	22	0b	Fatal error detected in IPSec post processing Rx FIFO
ghost_parity_desc_rd_fatal	23	0b	Fatal error detected in descriptor completion buffer
ghost_parity_data_rd_fatal	24	0b	Fatal error detected in data completion buffer
drx_ecc_icache_fatal	25	0b	Fatal error detected in Rx descriptor handler icache
dtx_ecc_icache_fatal	26	0b	Fatal error detected in Tx descriptor handler icache
dbu_ecc_sw_pb_fatal	27	0b	Fatal error detected in the header part of a packet in the switch packet buffer memory
dbu_ecc_tx_pb_fatal	28	0b	Fatal error detected in the header part of a packet in the Tx packet buffer memory
dbu_ecc_rx_pb_fatal	29	0b	Fatal error detected in the header part of a packet in the Rx packet buffer memory
mac_ecc_secu_tx_pb_fatal	30	0b	Fatal error detected in the Tx IPSec packet buffer memory
Mem_fault_port_hang (RO)	31	0b	Indicates a hang condition due to a fatal parity or ECC error in one of the memories.

8.23.34 Parity and ECC Indication Mask – PEINDM (0x1088; RW)

Field	Bit(s)	Initial Value	Description
dbu_ecc_sw_pb_nfatal	0	0b	Enable impact of error detected in the data part of a packet in the switch packet buffer memory.
dbu_ecc_tx_pb_nfatal	1	0b	Enable impact of error detected in the data part of a packet in the Tx packet buffer memory.
dbu_ecc_rx_pb_nfatal	2	0b	Enable impact of error detected in the data part of a packet in the Rx packet buffer memory.
Reserved	7:3	0x0	Reserved
dtx_parity_temp_fatal	8	0b	Enable impact of error detected in TSO prototype header memory.
mac_parity_secu_rx_sa_fatal	9	0b	Enable impact of error detected in IPSec Rx keys memory.
mac_parity_secu_tx_sa_fatal	10	0b	Enable impact of error detected in IPSec Tx keys memory.
drx_parity_desc_fifo_fatal	11	0b	Enable impact of error detected in internal received packets descriptor memory.
dtx_parity_dpt2_fatal	12	0b	Enable impact of error detected in internal transmit packets descriptor memory.
dtx_parity_hdr_fatal	13	0b	Enable impact of error detected in internal transmit packets descriptor memory.
dbu_parity_drx_pb_reg_fatal	14	0b	Enable impact of error detected in Rx packet buffer registers file.
dtx_parity_dh_reg_fatal	15	0b	Enable impact of error detected in Tx descriptor handler registers file.
drx_parity_dh_reg_fatal	16	0b	Enable impact of error detected in Rx descriptor handler registers file.
dtx_parity_lso_fatal	17	0b	Enable impact of error detected in TSO internal Tx context memory.
dtx_parity_cntxt_fatal	18	0b	Enable impact of error detected in internal Tx context memory.
xtx_parity_buf_stg_fatal	19	0b	Enable impact of error detected in XTX internal FIFO.
mac_parity_secu_post_l4cs_fatal	20	0b	Enable impact of error detected in IPSec Checksum Rx FIFO.
mac_parity_secu_pre_hdr_parse_fatal	21	0b	Enable impact of error detected in IPSec pre-processing Rx FIFO.
mac_parity_secu_post_sta_fatal	22	0b	Enable impact of error detected in IPSec post processing Rx FIFO.
ghost_parity_desc_rd_fatal	23	0b	Enable impact of error detected in descriptor completion buffer.
ghost_parity_data_rd_fatal	24	0b	Enable impact of error detected in data completion buffer.
drx_ecc_icache_fatal	25	0b	Enable impact of error detected in Rx descriptor handler icache.
dtx_ecc_icache_fatal	26	0b	Enable impact of error detected in Tx descriptor handler icache.
dbu_ecc_sw_pb_fatal	27	0b	Enable impact of error detected in the header part of a packet in the switch packet buffer memory.
dbu_ecc_tx_pb_fatal	28	0b	Enable impact of error detected in the header part of a packet in the Tx packet buffer memory.
dbu_ecc_rx_pb_fatal	29	0b	Enable impact of error detected in the header part of a packet in the Rx packet buffer memory.
mac_ecc_secu_tx_pb_fatal	30	0b	Enable impact of error detected in the Tx IPSec packet buffer memory.
Parity Enable	31	1b	Enable parity in all the parity protected memories.

8.23.35 Tx DMA Performance Burst and Descriptor Count - TXBDC (0x35E0; RC)

Field	Bit(s)	Initial Value	Description
BurstCNT	15:0	0x0	The counter is counting the transitions from Idle to burst state as long as the internal TXCYCLE counter does not equal 0xFFFF. The counter is clear on read.
DescCNT	31:16	0x0	The counter is counting the number of descriptors fetched as long as the internal TXCYCLE counter does not equal 0xFFFF. The counter is clear on read.

8.23.36 Tx DMA Performance Idle Count - TXIDLE (0x35E4; RC)

Field	Bit(s)	Initial Value	Description
SingleCNT	15:0	0x0	The counter is counting the transitions from Idle to single state as long as the internal TXCYCLE counter does not equal 0xFFFF. The counter is clear on read.
IdleCNT	31:16	0x0	The counter is counting the cycles on which the TxDMA is in idle for more then one cycle as long as the internal TXCYCLE counter does not equal 0xFFFF. The counter is clear on read.

Note: Reading this register will clear this internal TXCYCLE counter. This counter will then count for a period of 64K cycles. When this period is over the counters in TXBDC and this registers will freeze. In order to read the results of the measurements, the TXBDC register must be read first.

8.23.37 Rx DMA Performance Burst and Descriptor Count - RXBDC (0x25E0; RC)

Field	Bit(s)	Initial Value	Description
BurstCNT	15:0	0x0	The counter is counting the transitions from Idle to burst state as long as the internal RXCYCLE counter does not equal 0xFFFF. The counter is clear on read.
DescCNT	31:16	0x0	The counter is counting the number of descriptors fetched as long as the internal RXCYCLE counter does not equal 0xFFFF. The counter is clear on read.

8.23.38 Rx DMA Performance Idle Count - RXIDLE (0x25E4; RC)

Field	Bit(s)	Initial Value	Description
SingleCNT	15:0	0x0	The counter is counting the transitions from Idle to single state as long as the internal RXCYCLE counter does not equal 0xFFFF. The counter is clear on read.
IdleCNT	31:16	0x0	The counter is counting the cycles on which the TxDMA is in idle for more then one cycle as long as the internal RXCYCLE counter does not equal 0xFFFF. The counter is clear on read.

Note: Reading this register will clear this internal RXCYCLE counter. This counter will then count for a period of 64K cycles. When this period is over the counters in RXBDC and this registers will freeze. In order to read the results of the measurements, the RXBDC register must be read first.

8.24 PHY Software Interface (PHYREG)

- Base Registers (0 through 10 and 15) are defined in accordance with the “Reconciliation Sub layer and Media Independent Interface” and “Physical Layer Link Signaling for 10/100/ 1000 Mb/s Auto-Negotiation” sections of the IEEE 802.3.
- Additional registers (PHYREG.16 through 28) are defined in accordance with the IEEE 802.3 specification for adding unique chip functions.

Note: The PHY register bit descriptions follow Table .

PHYREG 26 is defined as a “secure register”. This means that software attempts to access this register is blocked by MAC, only internal hardware accesses (for example, firmware) are enabled.

Table 8-27. Table of PHYREG Registers

Offset	Abbreviation	Name	RW	Link to Page
00d	PCTRL	PHY Control Register	R/W	page 527
01d	PSTATUS	PHY Status Register	R	page 528
02d	PHY ID 1	PHY Identifier Register 1 (LSB)	R	page 528
03d	PHY ID 2	PHY Identifier Register 2 (MSB)	R	page 529
04d	ANA	Auto–Negotiation Advertisement Register	R/W	page 529
05d		Auto–Negotiation Base Page Ability Register	R	page 530
06d	ANE	Auto–Negotiation Expansion Register	R	page 530
07d	NPT	Auto–Negotiation Next Page Transmit Register	R/W	page 531
08d	LPN	Auto–Negotiation Next Page Ability Register	R	page 531
09d	GCON	1000BASE–T/100BASE–T2 Control Register	R/W	page 532
10d	GSTATUS	1000BASE–T/100BASE–T2 Status Register	R	page 532
15d	ESTATUS	Extended Status Register	R	page 533
16d	PCONF	Port Configuration Register	R/W	page 533
17d	PSTAT	Port Status 1 Register	RO	page 534
18d	PCONT	Port Control Register	RO	page 535
19d	LINK	Link Health Register	RO	page 536
20d	PFIFO	1000Base–T FIFO Register	R/W	page 537
21d	CHAN	Channel Quality Register	RO	page 537
25d		PHY Power Management	R/W	page 537
26d		Special Gigabit Disable Register	R/W	page 538
27d		Misc. Control Register 1	R/W	page 538
28d		Misc. Control Register 2	RO	page 539
31d		Page Select Core Register	WO	page 539

8.24.1 PHY Control Register - PCTRL (00d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	5:0	Reserved Always read as 0b. Write to 0b for normal operation	RW	Always 000000b
Speed Selection 1000 Mb/s (MSB)	6	Speed Selection is determined by bits 6 (MSB) and 13 (LSB) as follows. 11b = Reserved 10b = 1000 Mb/s 01b = 100 Mb/s 00b = 10 Mb/s 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	00b
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
Duplex Mode	8	1b = Full Duplex. 0b = Half Duplex. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	1b
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
Isolate	10	This bit has no effect on PHY functionality. Program to 0b for future compatibility.	R/W	0b
Power Down	11	1b = Power down. 0b = Normal operation. When using this bit, PHY default configuration is lost and is not loaded from the EEPROM after de-asserting the *Power Down* bit. Note: After this bit is set, all indications from PHY including link status are invalid.	R/W	0b
Auto-Negotiation Enable	12	1b = Enable Auto-Negotiation Process. 0b = Disable Auto-Negotiation Process. This bit must be enabled for 1000BASE-T operation.	R/W	1b
Speed Selection (LSB)	13	See Speed Selection (MSB), bit 6. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	1b
Loopback	14	1b = Enable loopback. 0b = Disable loopback.	R/W	0b
Reset	15	1b = PHY reset. 0b = Normal operation. **Note:** When using PHY Reset, the PHY default configuration is not loaded from the EEPROM. The preferred way to reset the 82576 PHY is using the CTRL.PHY_RST field.	WO, SC	0b

8.24.2 PHY Status Register - PSTATUS (01d; R)

Field	Bit(s)	Description	Mode	Default
Extended Capability	0	1b = Extended register capabilities.	RO	1b
Jabber Detect	1	1b = Jabber condition detected. 0b = Jabber condition not detected.	RO LH	0b
Link Status	2	1b = Link is up. 0b = Link is down.	RO, LL	0b
Auto-Negotiation Ability	3	1b = PHY able to perform Auto-Negotiation. 0b = PHY is not able to perform Auto-Negotiation.	RO	1b
Remote Fault	4	1b = Remote fault condition detected. 0b = Remote fault condition not detected.	RO LH	0b
Auto-Negotiation Complete	5	1b = Auto-Negotiation process complete. 0b = Auto-Negotiation process not complete.	RO	0b
MF Preamble Suppression	6	1b = PHY accepts management frames with preamble suppressed. 0b = PHY does not accept management frames with preamble suppressed.	RO	0b
Reserved	7	Reserved. Ignore on reads.	RO	0b
Extended Status	8	1b = Extended status information in the Extended PHY Status Register (15d). 0b = No extended status information in the Extended PHY Status Register (15d).	RO	1b
100BASE-T2 Half Duplex	9	1b = PHY able to perform half duplex 100BASE-T2 (not supported). 0b = PHY not able to perform half duplex 100BASE-T2.	RO	0b
100BASE-T2 Full Duplex	10	1b = PHY able to perform full duplex 100BASE-T2 (not supported). 0b = PHY not able to perform full duplex 100BASE-T2.	RO	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 not 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	1b = PHY able to perform 100BASE-T4. 0b = PHY not able to perform 100BASE-T4.	RO	0b

8.24.3 PHY Identifier Register 1 (LSB) - PHY ID 1 (02d; R)

Field	Bit(s)	Description	Mode	Default
PHY ID Number	15:0	The PHY identifier composed of bits 3 through 18 of the Organizationally Unique Identifier (OUI)	RO	0x02A8

8.24.4 PHY Identifier Register 2 (MSB) - PHY ID 2 (03d; R)

Field	Bit(s)	Description	Mode	Default
Manufacturer's Revision Number	3:0	4 bits containing the manufacturer's revision number.	RO	0x1
Manufacturer's Model Number	9:4	6 bits containing the manufacturer's part number.	RO	0x39
PHY ID Number	15:10	The PHY identifier composed of bits 19 through 24 of the OUI	RO	0x00

8.24.5 Auto–Negotiation Advertisement Register - ANA (04d; R/W)

Field	Bit(s)	Description	Mode	Default
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	00001b
10Base-T	5	1b = DTE is 10BASE-T capable. 0b = DTE is not 10BASE-T capable.	R/W	1b
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
100Base-TX	7	1b = DTE is 100BASE-TX capable. 0b = DTE is not 100BASE-TX capable.	R/W	1b[1]
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[1]
100BASE-T4	9	1b = Capable of 100BASE-T4 (not supported). 0b = Not capable of 100BASE-T4.	R/W	0b
PAUSE	10	Advertise to Partner that Pause operation (as defined in 802.3x) is desired.	R/W	1b
ASM_DIR	11	Advertise Asymmetric Pause direction bit. This bit is used in conjunction with PAUSE.	R/W	1b
Reserved	12	Always read as 0b. Write to 0b for normal operation.	R/W	0b
Remote Fault	13	1b = Set Remote Fault bit. 0b = Do not set Remote Fault bit.	R/W	0b
Reserved	14	Always read as 0b. Write to 0b for normal operation.	R/W	0b
Next Page	15	1b = Manual control of Next Page (Software). 0b = the 82576 control of Next Page (Auto).	R/W	0b

[1] If EEPROM ADV10LU (word 0x21, bit 3) is asserted, then the default is set to 0b; otherwise, the default is 1b.

8.24.6 Auto–Negotiation Base Page Ability Register - (05d; R)

Field	Bit(s)	Description	Mode	Default
Selector Fields[4:0]	4:0	<00001> = IEEE 802.3 Other combinations are reserved. Unspecified or reserved combinations must 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	10	Link Partner uses Pause Operation as defined in 802.3x.	RO	N/A
LP ASM_DIR	11	Asymmetric Pause Direction Bit 1b = Link Partner is capable of asymmetric pause. 0b = Link Partner is not capable of asymmetric pause.	RO	N/A
Reserved	12	Always read as 0b. Write as 0b.	RO	0b
Remote Fault	13	1b = Remote fault. 0b = No remote fault.	RO	N/A
Acknowledge	14	1b = Link Partner has received Link Code Word from the PHY. 0b = Link Partner has not received Link Code Word from the PHY.	RO	N/A
Next Page	15	1b = Link Partner has ability to send multiple pages. 0b = Link Partner has no ability to send multiple pages.	RO	N/A

8.24.7 Auto–Negotiation Expansion Register - ANE (06d; R)

Field	Bit(s)	Description	Mode	Default
Link Partner Auto-Negotiation Able	0	1b = Link Partner is Auto-Negotiation able. 0b = Link Partner is not Auto-Negotiation able.	RO	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
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

Field	Bit(s)	Description	Mode	Default
Parallel Detection Fault	4	1b = Parallel detection fault has occurred. 0b = Parallel detection fault has not occurred.	RO/LH	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.	RO/LH	0b
Reserved	15:6	Always read as 0b.	RO	0x0

8.24.8 Auto–Negotiation Next Page Transmit Register - NPT (07d; R/W)

Field	Bit(s)	Description	Mode	Default
Message/Unformatted Field	10:0	11-bit message code field.	R/W	0x1
Toggle	11	1b = Previous value of the transmitted Link Code Word = 0b. 0b = Previous value of the transmitted Link Code Word = 1b.	RO	0b
Acknowledge 2	12	1b = Complies with message. 0b = Cannot comply with message.	R/W	0b
Message Page	13	1b = Message page. 0b = Unformatted page.	R/W	1b
Reserved	14	Always read as 0b. Write to 0b for normal operation.	RO	0b
Next Page	15	1b = Additional next pages follow. 0b = Last page.	R/W	0b

8.24.9 Auto–Negotiation Next Page Ability Register - LPN (08d; R)

Bit(s)	Field	Description	Mode	Default
10:0	Message/Unformatted Field	11-bit message code field.	RO	0x0
11	Toggle	1b = Previous value of the transmitted Link Code Word = 0b. 0b = Previous value of the transmitted Link Code Word = 1b.	RO	0b
12	Acknowledge 2	1b = Link Partner complies with the message. 0b = Link Partner cannot comply with the message.	RO	0b
13	Message Page	1b = Page sent by the Link Partner is a Message Page. 0b = Page sent by the Link Partner is an Unformatted Page.	RO	0b
14	Acknowledge	1b = Link Partner has received Link Code Word from the PHY. 0b = Link Partner has not received Link Code Word from the PHY.	RO	0b
15	Next Page	1b = Link Partner has additional next pages to send. 0b = Link Partner has no additional next pages to send.	RO	0b

8.24.10 1000BASE–T/100BASE–T2 Control Register - GCON (09d; R/W)

Bit(s)	Field	Description	Mode	Default
7:0	Reserved	Always read as 0b. Write to 0b for normal operation.	R/W	0b
8	1000BASE-T Half Duplex	1b = DTE is 1000BASE-T capable. 0b = DTE is not 1000BASE-T capable. This bit is used by Smart Negotiation.	R/W	0b
9[1]	1000BASE-T Full Duplex	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
10	Port Type	1b = Prefer multi-port device (Master). 0b = Prefer single port device (Slave). This bit is only used when PHY register 9, bit 12 is set to 0b.	R/W	0b
11	Master/Slave Config Value	1b = Configure PHY as MASTER during MASTER-SLAVE negotiation (only when PHY register 9, bit 12 is set to 1b. 0b = Configure PHY as SLAVE during MASTER-SLAVE negotiation (only when PHY register 9, bit 12 is set to 1b.	R/W	0b
12	Master/Slave Config Enable	1b = Manual Master/Slave configuration. 0b = Automatic Master/Slave configuration.	R/W	0b
15:13	Test mode	000b = Normal Mode. 001b = Pulse and Droop Template. 010b = Jitter Template. 011b = Jitter Template. 100b = Distortion Packet. 101b, 110b, 111b = Reserved.	R/W	000b

[1] The default of this bit is affected by the EEPROM bit configuration of the 82576the 82575.
If EEPPROM bit AN-1000DIS is asserted, then the default is set to 0b.
If EEPPROM bit ADV10LU (word 0x21, bit 3) is asserted, then the default is set to 0b

8.24.11 1000BASE–T/100BASE–T2 Status Register - GSTATUS (10d; R)

Field	Bit(s)	Description	Mode
Idle Error Count	7:0	Idle Error counter Value. This register counts the number of invalid idle codes when link is high and the PHY is in either 1000BASE-T or 100BASE-T modes. If there is an overflow, these bits are all held at 1b. They are cleared on read or a hard or soft reset.	RO, LH
Reserved	9:8	Reserved. Always set to 00b.	RO
LP 1000T HD	10	1b = Link Partner is capable of 1000BASE-T half duplex. 0b = Link Partner is not capable of 1000BASE-T half duplex. Value in bit 10 are not valid until the ANE Register Page Received bit equals 1b.	RO
LP 1000T FD	11	1b = Link Partner is capable of 1000BASE-T full duplex. 0b = Link Partner is not capable of 1000BASE-T full duplex. Value in bit 11 are not valid until the ANE Register Page Received bit equals 1b.	RO
Remote Receiver Status	12	1b = Remote Receiver OK. 0 b = Remote Receiver Not OK.	RO
Local Receiver Status	13	1b = Local Receiver OK. 0b = Local Receiver Not OK.	RO
Master/Slave Resolution	14	1b = Local PHY configuration resolved to Master. 0b = Local PHY configuration resolved to Slave. Value in bits 14 are not valid until the ANE Register Page Received bit equals 1b.	RO
Master/Slave Config Fault	15	1b = Master/Slave configuration fault detected. 0b = No Master/Slave configuration fault detected.	RO, LH

8.24.12 Extended Status Register - ESTATUS (15d; R)

Field	Bit(s)	Description	Mode	Default
Reserved	11:0	Reserved. Always read as 0b.	RO	0x0
1000BASE-T Half Duplex	12	1b = 1000BASE-T half duplex capable. 0b = not 1000BASE-T half duplex capable.	RO	1b
1000BASE-T Full Duplex	13	1b = 1000BASE-T full duplex capable. 0b = Not 1000BASE-T full duplex capable.	RO	1b
1000BASE-X Half Duplex	14	1b =1000BASE-X half duplex capable. 0b = Not 1000BASE-X half duplex capable.	RO	0b
1000BASE-X Full Duplex	15	1b =1000BASE-X full duplex capable. 0b = Not 1000BASE-X full duplex capable.	RO	0b

8.24.13 Port Configuration Register - PCONF (16d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	0	Always read as 0b. Write to 0b for normal operation.	R/W	0b
Alternate NP Feature	1	1b = Enable alternate Auto-Negotiate next page feature. 0b = Disable alternate Auto-Negotiate next page feature.	R/W	0b
Reserved	3:2	Always read as 00b. Write to 00b for normal operation.	R/W	00b
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
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
Reserved	6	Always read as 0b. Write to 0b for normal operation.	R/W	0b
Smart Speed	7	1b = Smart Speed selection enabled. 0b = Smart Speed selection disabled. **Note:** The default of this bit is determined by the EEPROM speed bit (word 0x21, bit 5).	R/W	0b
TP Loopback (10BASE-T)	8	1b = Disable TP loopback during half-duplex operation. 0b = Normal operation.	R/W	1b
Reserved	9	Always read as 0b. Write to 0b for normal operation.	R/W	0b
Jabber (10BASE-T)	10	1b = Disable jabber. 0b = Enable jabber.	R/W	0b
Bypass 4B5B (100BASE-TX)	11	1b = Bypass4B5B encoder and decoder. 0b = Normal operation.	R/W	0b
Bypass Scramble (100BASE-TX)	12	1b = Bypass scrambler and descrambler. 0b = Normal operation.	R/W	0b

Field	Bit(s)	Description	Mode	Default
Transmit Disable	13	1b = Disable twisted-pair transmitter. 0b = Normal operation.	R/W	0b
Link Disable	14	1b = Force link pass 0b = Normal operation For 10BASE-T, this bit forces the link signals to be active. In 100BASE-T mode, setting this bit should force the Link Monitor into it's LINKGOOD state. For Gigabit operation, this merely bypasses Auto-Negotiation—the link signals still correctly indicate the appropriate status.	R/W	0b
Reserved	15	Always read as 0b. Write 0b for normal operation.	R/W	0b

8.24.14 Port Status 1 Register - PSTAT (17d; RO)

Field	Bit(s)	Description	Mode	Default
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
Polarity Status	1	1b = 10BASE-T polarity is reversed. 0b = 10BASE-T polarity is normal.	RO	0b
Reserved	6:2	Ignore these bits.	RO	0b
Reserved	8:7	Reserved Ignore these bits.	RO	X
Duplex Mode	9	1b = Full duplex. 0b = Half duplex.	RO	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
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
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
Transmit Status	13	1b = PHY currently transmitting a packet. 0b = PHY transmitter is IDLE. When in internal loopback, this bit reads as 0b.	RO	0b
Data Rate	15:14	00b = Reserved. 01b = PHY operating in 10BASE-T mode. 10b = PHY operating in 100BASE-TX mode. 11b = PHY operating in 1000BASE-T mode.	RO	00b

8.24.15 Port Control Register - PCONT (18d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	3:0	Always read as 0b. Write to 0b for normal operation.	R/W	0x0
TP Loopback	4	Allow gigabit loopback on twisted pairs.	R/W	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
Reserved	8:6	Always read as 0b. Write to 0b for normal operation.	R/W	0x0
Non-Compliant Scrambler Compensation	9	1b = Detect and correct for non-compliant scrambler. 0b = Detect and report non-compliant scrambler. **Note:** The default of this bit is affected by the EEPROM bit configurations of the 82576the 82575. If EEPROM word 0x21, bit 2 is asserted, then the default is set to 1b.	R/W	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
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
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
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
Reserved	14	Always read as 0b. Write to 0b for normal operation.	R/W	0b
Jitter Test Clock	15	This configuration bit is used to enable the 82576the 82575 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 0x4011, 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

8.24.16 Link Health Register - LINK (19d; RO)

Field	Bit(s)	Description	Mode	HW Rst
Valid Channel A	0	The channel A DSP had converged to incoming data.	RO	0b
Valid Channel B	1	The channel B DSP had converged to incoming data.	RO	0b
Valid Channel C	2	The channel C DSP had converged to incoming data.	RO	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
Auto-Negotiation Active	4	Auto-Negotiate is actively deciding HCD.	RO	0b
Reserved	5	Always read as 0b.	RO	0b
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
Reserved	7	Always read as 0b.	RO	0b
Data Err[0]	8	Mode: 10: 10 Mb/s polarity error. 100: Symbol error. 1000: Gig idle error.	RO/LH	X
Data Err[1]	9	Mode: 10: N/A. 100: Scrambler unlocked. 1000: Local receiver not OK.	RO/LH	X
Count Overflow	10	32 idle error events were counted in less than 1 ms.	RO/LH	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
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
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
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
SS Downgrade	15	Smart Speed has downgraded the link speed from the maximum advertised.	RO/LH	0b

8.24.17 1000Base–T FIFO Register - PFIFO (20d; R/W)

Field	Bit(s)	Description	Mode	Default
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 0xD.	R/W	0101b
Reserved	7:4	Must be set to 0100 for normal operation.	R/W	0100b
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
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
Reserved	13:11	Always read as 0b. Write to 0b for normal operation.	R/W	0x0
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
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

8.24.18 Channel Quality Register - CHAN (21d; RO)

Field	Bit(s)	Description	Mode	Default
MSE_A	3:0	The converged mean square error for Channel A.	RO	0x0
MSE_B	7:4	The converged mean square error for Channel B.	RO	0x0
MSE_C	11:8	The converged mean square error for Channel C.	RO	0x0
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	0x0

8.24.19 PHY Power Management - (25d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	15:9	Always read as 0b. Write to 0b for normal operation.	R/W	0x0
rst_compl	8	Indicates PHY internal reset cleared.	LH	0b
SPD_B2B_EN	7	SPD back-to-back enable.	R/W	1b
Disable 1000	6	When set, disables 1000 Mb/s in all power modes. Note that this bit can be loaded from EEPROM.	R/W	0b
Go Link disconnect	5	Setting this bit will cause the PHY to enter link disconnect mode immediately.	R/W	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
Disable 1000 nD0a	3	Disables 1000 Mb/s operation in non-D0a states. Note that this bit can be loaded from EEPROM.	R/W	1b
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. If this bit is loaded from EEPROM, it is reset to the EEPROM value after each fundamental reset.	R/W	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
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

Note: Part of the default values of this register can be changed by EEPROM settings.

8.24.20 Special Gigabit Disable Register - (26d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	15:0	Always read as 0b. Write to 0b for normal operation.	R/W	0x0

8.24.21 Misc. Control Register 1 - (27d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	15	Ignore this bit.	R/W	0b
Reserved	14	Always read as 0b. Write to 0b for normal operation.	RO	0b
Reserved	13	Must be 1 for normal operation.	R/W	1b
Duplex_manual_set	12	When set, the 82576 sets the duplex according to the Duplex Mode bit in register 0. This bit is cleared following auto-negotiation.	R/W	0b
Reserved	11:9	Always read as 0b. Write to 0b for normal operation.	RO	0x0
ss_cfg_cntr	8:6	Smart speed counter configuration: 1-5 (001b:101b).	R/W	010b
T10_auto_pol_dis	5	When set, disables the auto-polarity mechanism in the 10 block.	R/W	0b
Reserved	4:0	Always read as 0b. Write to 0b for normal operation.	R/W	0x0

8.24.22 Misc. Control Register 2 - (28d; RO)

Field	Bit(s)	Description	Mode	Default
Reserved	15:14	Always read as 0b. Write to 0b for normal operation.	R/W	0x0
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
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
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
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
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
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
Reserved	7:0	Reserved.	R/W	0x0

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).

8.24.23 Page Select Core Register - (31d; WO)

Field	Bit(s)	Description	Mode	Default
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	0x0

8.25 Virtual Function Device registers

8.25.1 Queues Registers

Each VF has two queues – Q0 and Q1. These queues are also used by the PF when working in non-IOV mode or if not all VFs are allocated to VMs. The mapping between the Virtual Queue number (VQn) and the Physical Queue number (PQn) is given by the equation: **PQn = VFn + VQn*8** (where VFn is the VF number).

For example: Q0 and Q1 of VF0 are actually Q0 and Q8, Q0 and Q1 of VF1 are actually Q1 and Q9, etc.

The virtual address of Q0 and Q1 registers is always the same (RX: 0x2800 and 0x2880, TX: 0x3800 and 0x3880) – like the physical Q0 and Q1 in the 82575 aliased area.

8.25.2 Non-queue Registers

Non-queue registers get a virtual address that are equal to the same registers that belong to the PF. These registers are mapped to the physical address space at 0x10000, where each VM gets 0x100 bytes for its registers:

- VF0 registers: 0x10000 – 0x100FF
- VF1 registers: 0x10100 – 0x101FF
- ...

8.25.2.1 EITR registers

The 82576 supports 25 EITR registers. In non IOV mode, all the EITR registers can be used by the PF. In IOV mode, 3 EITR registers are allocated to each VF and the PF should only use the remaining EITR registers. EITR0-2 registers are accessed by the VFs at addresses 0x1680 - 0x1688 and matches the PF EITRs according to the following table. EITR0 is always allocated to the PF.

VF	PF EITR	Physical Address
0	EITR22 - EITR24	0x16D8 - 0x16E0
1	EITR19 - EITR21	0x16CC - 0x16D4
2	EITR16 - EITR18	0x16C0 - 0x16C8
...		
7	EITR1 - EITR3	0x1684 - 0x168C

8.25.2.2 MSI-X registers

The MSI-X vectors of each VF are reflected in it's BAR3.

The PBA bits of the VFs are not replicated in the PF.

8.25.3 Register set - CSR BAR

Virtual Address[1]	Physical Address Base	Abbreviation	Name
0x0000/0x0004	0x10000 + VFn * 0x100	VTCTRL	Control (only RST bit)
0x0008	0x0008 (Common – RO)	VTStatus	Status (mirror of PF status register).
0x1048	0x1048 (common - RO)	VTFRTIMER	Free running timer (mirror of PF timer).
0x1520	0x10020 + VFn * 0x100	VTEICS	Extended Interrupt Cause Set Register
0x1524	0x10024 + VFn * 0x100	VTEIMS	Extended Interrupt Mask Set/Read Register
0x1528	0x10028 + VFn * 0x100	VTEIMC	Extended Interrupt Mask Clear Register
0x152C	0x1002C + VFn * 0x100	VTEIAC	Extended Interrupt Auto Clear Register
0x1530	0x10030 + VFn * 0x100	VTEIAM	Extended Interrupt Auto Mask Enable register
0x1580	0x10080 + VFn * 0x100	VTEICR	Extended Interrupt Cause Set Register

0x1680 – 0x1688	0x16D8 - 0x16E0 - VFn * 0xC	EITR 0-2	Interrupt Throttle Registers 0-2
0x1700	0x1700 + VFn * 4	VTIVAR0	Interrupt vector allocation register Queues
0x1740	0x1720 + VFn * 4	VTIVAR_MISC	Interrupt vector allocation register Misc.
0x0F04	0x5B68	PBACL	PBA clear
0x0F0C	0x5480 + VFn * 0x4	PSRTYPE	Replication Packet Split Receive Type
0x0C40	0x0C40 + VFn*0x4	VFMailbox	Virtual Function Mailbox
0x0800 – 0x083F	0x0800 – 0x083F + VFn * 0x40	VMBMEM	Virtualization Mail Box Memory
0x2800+n*0x100 n=0,1	0xC000, 0xC200 +VFn * 0x40	RDBAL0/1	Receive Descriptor Base Address Low 0/1
0x2804+n*0x100 n=0,1	0xC004, 0xC204 +VFn * 0x40	RDBAH0/1	Receive Descriptor Base Address High 0/1
0x2808+n*0x100 n=0,1	0xC008, 0xC208 +VFn * 0x40	RDLEN0/1	Receive Descriptor Length 0/1
0x280C+n*0x100 n=0,1	0xC00C, 0xC20C +VFn * 0x40	SRRCTL0/1	Split and Replication Receive Control Register 0/1
0x2810+n*0x100 n=0,1	0xC010, 0xC210 +VFn * 0x40	RDH0/1	Receive Descriptor Head 0/1
0x2814+n*0x100 n=0,1	0xC014, 0xC214 +VFn * 0x40	RXCTL0/1	Rx DCA control registers 0/1
0x2818+n*0x100 n=0,1	0xC018, 0xC218 +VFn * 0x40	RDT0/1	Receive Descriptor Tail 0/1
0x2828+n*0x100 n=0,1	0xC028, 0xC228 +VFn * 0x40	RXDCTL0/1	Receive Descriptor Control 0/1
0x2830+n*0x100 n=0,1	0xC030, 0xC230 +VFn * 0x40	RQDPC0/1	Receive Queue drop packet count 0/1
0x3800+n*0x100 n=0,1	0xE000, 0xE200 +VFn * 0x40	TDBAL0/1	Transmit Descriptor Base Address Low 0/1
0x3804+n*0x100 n=0,1	0xE004, 0xE204 +VFn * 0x40	TDBAH0/1	Transmit Descriptor Base Address High 0/1
0x3808+n*0x100 n=0,1	0xE008, 0xE208 +VFn * 0x40	TDLEN0/1	Transmit Descriptor Ring Length 0/1
0x3810+n*0x100 n=0,1	0xE010, 0xE210 +VFn * 0x40	TDH0/1	Transmit Descriptor Head 0/1
0x3814+n*0x100 n=0,1	0xE014, 0xE214 +VFn * 0x40	TXCTL0/1	Tx DCA control registers 0/1
0x3818+n*0x100 n=0,1	0xE018, 0xE218 +VFn * 0x40	TDT0/1	Transmit Descriptor Tail 0/1
0x3828+n*0x100 n=0,1	0xE028, 0xE228 +VFn * 0x40	TXDCTL0/1	Transmit Descriptor Control 0/1
0x3838+n*0x100 n=0,1	0xE038, 0xE238 +VFn * 0x40	TDWBAL0/1	Tx Descriptor Completion writeback Address Low 0/1
0x383C+n*0x100 n=0,1	0xE03C, 0xE23C +VFn * 0x40	TWBAH0/1	Tx Descriptor Completion writeback Address High 0/1
0x0F10	0x10010 + VFn * 0x100	VFGPRC	Good Packets Received Count
0x0F14	0x10014 + VFn * 0x100	VFGPTC	Good Packets Transmitted Count
0x0F18	0x10018 + VFn * 0x100	VFGORC	Good Octets Received Count
0x0F34	0x10034 + VFn * 0x100	VFGOTC	Good Octets Transmitted Count

0xF3C	0x1003C + VFn * 0x100	VFMPRC	Multicast Packets Received Count
0x0F40	0x10040 + VFn * 0x100	VFGPRLBC	Good RX Packets loopback Count
0x0F44	0x10044 + VFn * 0x100	VFGPTLBC	Good TX packets loopback Count
0x0F48	0x10048 + VFn * 0x100	VFGORLBC	Good RX Octets loopback Count
0x0F50	0x10050 + VFn * 0x100	VFGOTLBC	Good TX Octets loopback Count
0x34e8	0x34e8	PBTWAC	Tx packet buffer wrap around counter
0x24e8	0x24e8	PBRWAC	Rx packet buffer wrap around counter
0x30e8	0x30e8	PBSWAC	Switch packet buffer wrap around counter

1. Addresses in Red indicates registers whose virtual addresses are different from their physical addresses due to the need to maintain a virtual address space of 16KBytes.

8.25.4 Register set - MSI-X BAR

Virtual Address	Physical Address Base (+ VFn *0x30)	Abbreviation	Name
0x0000 - 0x0020	0x00010	MSIXTADD	MSIX table entry lower address
0x0004 - 0x0024	0x00018	MSIXTUADD	MSIX table entry upper address
0x0008 - 0x0028	0x00028	MSIXTMSG	MSIX table entry message
0x000C - 0x002C	N/A	MSIXTVCTRL	MSIX table vector control
Max(Page Size, 0x2000)	N/A	MSIXPBA	MSI-X Pending bit array

8.26 Virtual function Register Descriptions

All the registers in this section are replicated per VF. The addresses are relative to the beginning of each VF address space. The address relative to BAR0 as programmed in the IOV structure in the PF configuration space (offset 0x180-0x184) can be found by the following formula:

VF BAR0 + Max(16K, system page size)* VF# + CSR offset.

See Section 8.25.3 for the list of registers exposed to the VF detailed below.

8.26.1 VT control register - VTCTRL (0x0000; RW)

Field	Bit(s)	Initial Value	Description
Reserved	25:0	0x0	Reserved.
RST	26	0b	VF Reset This bit performs a reset of the queue enable and the interrupt registers of the VF.
Reserved	31:27	0x0	Reserved

8.26.2 VF Status Register - STATUS (0x00008; R)

This register is a mirror of the PF status register. See Table for details of this register.

8.26.3 VT Free Running Timer - VTFRTIMER (0x01048; RO)

This register reflects the value of a free running timer that can be used for various timeout indications. The register is reset by a PCI reset and/or software reset. This register is a mirror of the PF register. See description of this register in Section 8.16.3.

8.26.4 VT Extended Interrupt Cause - VTEICR (0x01580; RC/W1C)

See description of this register in Section 8.8.1.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Indicates an interrupt cause mapped to MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved

8.26.5 VT Extended Interrupt Cause Set - VTEICS (0x01520; WO)

See the description of this register in Section 8.8.2.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Sets to corresponding EICR bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved

8.26.6 VT Extended Interrupt Mask Set/Read - VTEIMS (0x01524; RWS)

See the description of this register in Section 8.8.3.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Set Mask bit for the corresponding EICR bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved

8.26.7 VT Extended Interrupt Mask Clear - VTEIMC (0x01528; WO)

See the description of this register in Section 8.8.4.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	clear Mask bit for the corresponding EICR bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved

8.26.8 VT Extended Interrupt Auto Clear - VTEIAC (0x0152C; R/W)

See the description of this register in Section 8.8.5.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Auto clear bit for the corresponding EICR bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved

8.26.9 VT Extended Interrupt Auto Mask Enable - VTEIAM (0x01530; R/W)

See the description of this register in Section 8.8.6.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Auto Mask bit for the corresponding EICR bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved

8.26.10 VT Interrupt Throttle - VTEITR (0x01680 + 4*n[n = 0...2]; R/W)

See the description of this register in Section 8.8.12.

Field	Bit(s)	Initial Value	Description
LLinterval	4:0	0x0	Low latency credits increment rate. The interval is specified in 8µs. Zero value gives the LL interrupt maximum bandwidth.
Interval	15:5	0x0	Minimum inter-interrupt interval. The interval is specified in 8 µs increments. A zero disables interrupt throttling logic. For Backward compatibility legacy driver writes to the LLinterval 0 then the interval can be interpreted as having a resolution of 256ns (using the LLinterval bits).
LL Counter (RWS)	20:16	0x0	Reflects the current credits for that EITR for LL interrupts.
Counter (RWS)	31:21	0x0	Down Counter Loaded with the interval value each time the associated interrupt is signaled. Counts down to 0 and stops. The associated interrupt is signaled each time this counter is zero and an associated (via the Interrupt Select register) EICR bit is set.

8.26.11 VT Interrupt Vector Allocation Registers - VTIVAR (0x01700; RW)

These registers define the allocation of the two queue pairs interrupt causes as defined in Table 7-44 to one of the MSI-X vectors. Each INT_Alloc[i] (i=0...3) field is a byte indexing an entry in the MSI-X Table Structure and MSI-X PBA Structure.

Field	Bit(s)	Initial Value	Description
INT_Alloc[0]	1:0	X	Defines the MSI-X vector assigned to the interrupt cause associated with queue 0 Rx. Valid values are 0 to 2.
Reserved	6:2	0x0	Reserved
INT_Alloc_val[0]	7	0b	Valid bit for INT_Alloc[0]
INT_Alloc[1]	9:8	X	Defines the MSI-X vector assigned to the interrupt cause associated with queue 0 Tx. Valid values are 0 to 2.
Reserved	14:10	0x0	Reserved
INT_Alloc_val[1]	15	0b	Valid bit for INT_Alloc[1]
INT_Alloc[2]	17:16	X	Defines the MSI-X vector assigned to the interrupt cause associated with queue 1 Rx. Valid values are 0 to 2.
Reserved	22:18	0x0	Reserved
INT_Alloc_val[2]	23	0b	Valid bit for INT_Alloc[2]

Field	Bit(s)	Initial Value	Description
INT_Alloc[3]	25:24	X	Defines the MSI-X vector assigned to the interrupt cause associated with queue 1 Tx. Valid values are 0 to 2.
Reserved	30:26	0x0	Reserved
INT_Alloc_val[3]	31	0b	Valid bit for INT_Alloc[3]

8.26.12 VT Interrupt Vector Allocation Registers - VTIVAR_MISC (0x01740; RW)

This register defines the MSI-X vector allocated to the mailbox interrupt.

A mailbox interrupt is asserted in the VF upon reception of a mailbox message or an acknowledge from the PF. It also asserted when the RSTI bit rises.

Field	Bit(s)	Initial Value	Description
INT_Alloc[4]	1:0	X	Defines the MSI-X vector assigned to the interrupt cause associated with the mailbox. Valid values are 0 to 2.
Reserved	6:2	0x0	Reserved
INT_Alloc_val[4]	7	0b	Valid bit for INT_Alloc[4]
Reserved	31:8	0x0	Reserved

8.26.13 MSI—X Table Entry Lower Address - MSIXTADD (BAR3: 0x0000 + 16*n [n=0...2]; R/W)

See Section 8.9.1 for information about this register.

8.26.14 MSI—X Table Entry Upper Address - MSIXTUADD (BAR3: 0x0004 + 16*n [n=0...2]; R/W)

See Section 8.9.2 for information about this register..

8.26.15 MSI—X Table Entry Message - MSIXTMSG (BAR3: 0x0008 + 16*n [n=0...2]; R/W)

See Section 8.9.3 for information about this register.

8.26.16 MSI—X Table Entry Vector Control - MSIXTVCTRL (BAR3: 0x000C + 16*n [n=0...2]; R/W)

See Section 8.9.4 for information about this register.

8.26.17 MSIXPBA - MSIXPBA Bit Description (BAR3: 0x02000; RO)

Field	Bit(s)	Initial Value	Description
Pending Bits	2:0	0x0	For each pending bit that is set, the function has a pending message for the associated MSI-X Table entry. Pending bits that have no associated MSI-X table entry are reserved.
Reserved	31:3	0x0	Reserved

Note: If a page size larger than 8K is programmed in the IOV structure, the address of the MSIX PBA table moves to be page aligned.

8.26.18 MSI—X PBA Clear - PBACL (0x00F04; R/W1C)

Field	Bit(s)	Initial Value	Description
PENBIT	2:0	0x0	MSI-X Pending bits Clear Writing a 1b to any bit clears the corresponding MSIXPBA bit; writing a 0b has no effect. Reading this register returns the PBA vector.
Reserved	31:3	0x0	Reserved

8.26.19 Receive Descriptor Base Address Low - RDBAL (0x02800 + 256*n [n=0...1];R/W)

See Section 8.10.5 for information about this register.

8.26.20 Receive Descriptor Base Address High - RDBAH (0x02804 + 256*n [n=0...1]; R/W)

See Section 8.10.6 for information about this register.

8.26.21 Receive Descriptor Ring Length - RDLEN (0x02808 + 256*n [n=0...1]; R/W)

See Section 8.10.7 for information about this register.

8.26.22 Receive Descriptor Head - RDH (0x02810 + 256*n [n=0...1]; R/W)

See Section 8.10.8 for information about this register.

8.26.23 Receive Descriptor Tail - RDT (0x02818 + 256*n [n=0...1]; R/W)

See Section 8.10.9 for information about this register.

8.26.24 Receive Descriptor Control - RXDCTL (0x02828 + 256*n [n=0...1]; R/W)

See Section 8.10.10 for information about this register.

8.26.25 Split and Replication Receive Control Register queue - SRRCTL (0x0280C + 256*n [n=0...1]; R/W)

See Section 8.10.2 for information about this register.

8.26.26 Receive Queue drop packet count - RQDPC (0x2830 + 256*n [n=0...1]; RC)

See Section 8.10.11 for information about this register.

8.26.27 Replication Packet Split Receive Type - PSRTYPE (0x00F0C; R/W)

See Section 8.10.3 for information about this register.

8.26.28 Transmit Descriptor Base Address Low - TDBAL (0x3800 + 256*n [n=0...1]; R/W)

See Section 8.12.8 for information about this register.

8.26.29 Transmit Descriptor Base Address High - TDBAH (0x03804 + 256*n [n=0...1]; R/W)

See Section 8.12.9 for information about this register.

8.26.30 Transmit Descriptor Ring Length - TDLEN (0x03808 + 256*n [n=0...1]; R/W)

See Section 8.12.10 for information about this register.

8.26.31 Transmit Descriptor Head - TDH (0x03810 + 256*n [n=0...1]; R/W)

See Section 8.12.11 for information about this register.

8.26.32 Transmit Descriptor Tail - TDT (0x03818 + 256*n [n=0...1]; R/W)

See Section 8.12.12 for information about this register.

8.26.33 Transmit Descriptor Control - TXDCTL (0x03828 + 256*n [n=0...1]; R/W)

See Section 8.12.13 for information about this register.

8.26.34 Tx Descriptor Completion Write–Back Address Low - TDWBAL (0x03838 + 256*n [n=0...1]; R/W)

See Table 8.12.14 for information about this register.

8.26.35 Tx Descriptor Completion Write–Back Address High - TDWBAH (0x0383C + 256*n [n=0...1];R/W)

See Section 8.12.15 for information about this register.

8.26.36 Rx DCA Control Registers - RXCTL (0x02814 + 256*n [n=0...1]; R/W)

See Section 8.13.1 for information about this register. for information about this register.

8.26.37 Tx DCA Control Registers - TXCTL (0x03814 + 256*n [n=0...1]; R/W)

See Section 8.13.2 for information about this register.

8.26.38 Good Packets Received Count - VFGPRC (0x0F10; RO)

This register counts the number of good packets received by the queues allocated to this VF of any legal length. This counter includes loopback packets or replications of multicast packets.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

An FLR to the VF may cause some inaccuracy in this counter.

Field	Bit(s)	Initial Value	Description
GPRC	31:0	0x0	Number of good packets received (of any length).

8.26.39 Good Packets Transmitted Count - VFGPTC (0x0F14; RO)

This register counts the number of good packets transmitted by the queues allocated to this VF. This counter includes loopback packets or packets latter dropped by the switch or the MAC.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GPTC	31:0	0x0	Number of good packets sent.

8.26.40 Good Octets Received Count - VFGORC (0x0F18; RO)

This register counts the number of good (no errors) octets received by this VF. This counter includes loopback packets or replications of multicast packets. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusive.

Only octets of packets that pass address filtering are counted in this register.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

An FLR to the VF may cause some inaccuracy in this counter.

Field	Bit(s)	Initial Value	Description
GORC	31:0	0x0	Number of good octets received

8.26.41 Good Octets Transmitted Count - VFGOTC (0x0F34; RO)

This register counts the number of good (no errors) octets transmitted by the queues allocated to this VF.

This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive, including any padding added by the hardware. The VLAN tag added by the hardware is counted as part of the packet. This register counts octets in successfully transmitted packets that are 64 or more bytes in length. This counter includes loopback packets or packets latter dropped by the switch or the MAC.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GOTC	31:0	0x0	Number of good octets transmitted

8.26.42 Multicast Packets Received Count - VFMPRC (0x0F3C; RO)

This register counts the number of good (no errors) multicast packets received by a given VM. 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.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0x0	Number of multicast packets received.

8.26.43 Good TX Octets loopback Count - VFGOTLBC (0x0F50; RO)

This register counts the number of good (no errors) octets transmitted by the queues allocated to this VF that where sent to local VF. This counter includes packets that are sent to the LAN and to a local VM.

This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive, including any padding added by the hardware. The VLAN tag added by the hardware is counted as part of the packet.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GOTLBC	31:0	0x0	Number of good octets transmitted to loopback

8.26.44 Good TX packets loopback Count - VFGPTLBC (0x0F44; RO)

This register counts the number of good (no errors) packets transmitted by the queues allocated to this VF that where sent to local VF. This counter includes packets that are sent to the LAN and to a local VM.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GPTLBC	31:0	0x0	Number of good packets transmitted to loopback

8.26.45 Good RX Octets loopback Count - VFGORLBC (0x0F48; RO)

This register counts the number of good (no errors) octets received by the queues allocated to this VF that where sent from some local VFs.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GORLBC	31:0	0x0	Number of good octets received from loopback

8.26.46 Good RX Packets loopback Count - VFGPRLBC (0x0F40; RO)

This register counts the number of good (no errors) packets received by the queues allocated to this VF that where sent from some local VFs.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GPRLBC	31:0	0x0	Number of good packets received from loopback

8.26.47 Virtual Function Mailbox - VFMailbox (0x0C40; RW)

See Section 8.14.3 for information about this register.

8.26.48 Virtualization Mailbox memory - VMBMEM (0x0800:0x083C; R/W)

A 64 bytes mailbox memory for PF and VF driver communication. Locations can be accessed as 32-bit or 64-bit words.

See Section 8.14.4 for information about this register.

8.26.49 Tx packet buffer wrap around counter - PBTWAC (0x34e8; RO)

See Section 8.3.4 for information about this register.

8.26.50 Rx packet buffer wrap around counter - PBRWAC (0x24e8; RO)

See Section 8.3.5 for information about this register.

8.26.51 Switch packet buffer wrap around counter - PBSWAC (0x30e8; RO)

See Section 8.3.6 for information about this register.

§ §

NOTE: ***This page intentionally left blank.***

9. PCIe Programming Interface

9.1 PCIe Compatibility

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

- All devices required to be supported by deployed PCI software must be enumerable as part of a tree through PCI device enumeration mechanisms.
- Devices in their default operating state must conform 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 capabilities 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 82576 is a multi-function device with the following functions:

Table 9-1. Intel® 82576 GbE Controller Functions

Function Number	Function Description	Disable options
0 or 1	LAN 0	Strapping option
1 or 0	LAN 1	Strapping option/EEPROM word 0x10, bit 11

LAN0 and LAN1 are shown in PCI functions 0 and 1. The *LAN Function Sel* field in EEPROM word 0x21 (reflected in the FACTPS register (0x5B30)) determines if LAN0 appears in PCI function 0 or PCI function 1. LAN1 appears in the complementary PCI function. See Section 4.3 for description of the functions mapping when part of the functions are disabled.

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

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

9.2 Configuration Sharing Among PCI Functions

The 82576 contains a single physical PCIe core interface. The 82576 is designed so that each of the logical LAN devices appears as a distinct function. Many of the fields of the PCIe header space contain hardware default values that are either fixed or might be overridden using EEPROM, but might not be independently specified for each logical LAN device. The following fields are considered to be common to both LAN devices:

Table 9-2. Common Fields for LAN Devices

Vendor ID	Fixed to 8086.
Revision	The revision number of the 82576 is reflected identically for both LAN functions.
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 functions, but the actual value reflected depends on LAN disable configuration. See Section 9.4.9 for details.
Subsystem ID	The subsystem ID of the 82576 can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for both LAN functions.
Subsystem Vendor ID	The subsystem vendor ID of the 82576 can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for both LAN functions.
Cap_Ptr, Max Latency, Min Grant	These fields reflect fixed values that are constant values reflected for both LAN functions.

The following fields are implemented individually for each LAN function:

Table 9-3. Fields Implemented Differently in LAN Functions

Device ID	The device ID reflected for each LAN function can be independently specified via EEPROM.
Command, Status	Each LAN function implements its own command/status registers.
Latency Timer, Cache Line Size	Each LAN function implements these registers individually. The system should program these fields identically for each LAN to ensure consistent behavior and performance of each function.
Memory BAR, Flash BAR, IO BAR, MSI-X BAR, Expansion ROM BAR,	Each LAN function implements its own base address registers, enabling each function to claim its own address region(s).
Interrupt Pin	Each LAN function independently indicates which interrupt pin (INTA# or INTB#) is used by that function's MAC to signal system interrupts. The value for each LAN function can be independently specified via EEPROM, but only if both LAN functions are enabled.
Class Code	Different class code values (iSCSI/LAN) can be set for each function.

See Section 9.7 for a description of the configuration space reflected to virtual functions.

9.3 Register Map

9.3.1 Register Attributes

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

Table 9-4. Configuration Registers

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-write and can be either set or reset.
R/W1C	Read-only status, write-1-to-clear status 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 via AUX power or PME enable) is enabled.
RWS	Read-write register: Register bits are read-write and can be either set or reset by software to the desired state. 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 via AUX power or PME enable) is enabled.

Table 9-4. Configuration Registers (Continued)

R/W1CS	Read-only status, write-1-to-clear status register: Register bits indicate status when read, a set bit indicating a status event can 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 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. Bits are read-only after initialization and can only be reset (for write-once by firmware) with PWRGOOD signal.
RsvdP	Reserved and preserved: Reserved for future R/W implementations; software must preserve value read for writes to bits.
RsvdZ	Reserved and zero: Reserved for future R/W1C implementations; software must use 0b for writes to bits.

The PCI configuration registers map is listed in Table 9-5. Refer to a detailed description for registers loaded from the EEPROM at initialization time. Note that initialization values of the configuration registers are marked in parenthesis.

9.3.2 PCIe Configuration Space Summary

Table 9-5. PCIe Configuration Registers Map

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
Mandatory PCI register	0x0	Device ID		Vendor ID	
	0x4	Status Register		Control Register	
	0x8	Class Code (0x020000/0x010000)			Revision ID
	0xC	BIST (0x00)	Header Type (0x0/ 0x80)	Latency Timer	Cache Line Size (0x10)
	0x10	Base Address Register 0			
	0x14	Base Address Register 1			
	0x18	Base Address Register 2			
	0x1C	Base Address Register 3			
	0x20	Base Address Register 4			
	0x24	Base Address Register 5			
	0x28	CardBus CIS pointer (0x0000)			
	0x2C	Subsystem Device ID		Subsystem Vendor ID	
	0x30	Expansion ROM Base Address			
	0x34	Reserved			Cap Ptr (0x40)
	0x38	Reserved			
	0x3C	Max Latency (0x00)	Min Grant (0x00)	Interrupt Pin (0x01/ 0x02)	Interrupt Line (0x00)
Power management capability	0x40	Power Management Capabilities		Next Pointer (0x50)	Capability ID (0x01)
	0x44	Data	Bridge Support Extensions	Power Management Control & Status	
MSI capability	0x50	Message Control (0x0080)		Next Pointer (0x70)	Capability ID (0x05)
	0x54	Message Address			
	0x58	Message Upper Address			
	0x5C	Reserved		Message Data	
	0x60	Mask bits			
	0x64	Pending bits			

Table 9-5. PCIe Configuration Registers Map (Continued)

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
MSI-X capability	0x70	Message Control (0x00090)		Next Pointer (0xA0)	Capability ID (0x11)
	0x74	Table Offset			
	0x78	PBA offset			
PCIe capability	0xA0	PCIe Capability Register (0x0002)		Next Pointer (0xE0)	Capability ID (0x10)
	0xA4	Device Capability			
	0xA8	Device Status		Device Control	
	0xAC	Link Capability			
	0xB0	Link Status		Link Control	
	0xB4	Reserved			
	0xB8	Reserved		Reserved	
	0xBC	Reserved			
	0xC0	Reserved		Reserved	
	0xC4	Device Capability 2			
	0xC8	Reserved		Device Control 2	
	0xCC	Reserved			
	0xD0	Reserved		Reserved	
	0xD4	Reserved			
	0xD8	Reserved		Reserved	
VPD capability	0xE0	VPD address		Next Pointer (0x00)	Capability ID (0x03)
	0xE4	VPD data			
AER capability	0x100	Next Capability Ptr. (0x140)	Version (0x1)	AER Capability ID (0x0001)	
	0x104	Uncorrectable Error Status			
	0x108	Uncorrectable Error Mask			
	0x10C	Uncorrectable Error Severity			
	0x110	Correctable Error Status			
	0x114	Correctable Error Mask			
	0x118	Advanced Error Capabilities and Control Register			
	0x11C: 0x128	Header Log			
Serial ID capability	0x140	Next Capability Ptr. (0x150)	Version (0x1)	Serial ID Capability ID (0x0003)	
	0x144	Serial Number Register (Lower Dword)			
	0x148	Serial Number Register (Upper Dword)			
ARI capability	0x150	Next Capability Ptr. (0x160)	Version (0x1)	ARI Capability ID (0x000E)	
	0x154	ARI Control Register		ARI Capabilities	

Table 9-5. PCIe Configuration Registers Map (Continued)

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
SR-IOV capability	0x160	Next Capability offset (0x0)	Version (0x1)	IOV Capability ID (0x0010)	
	0x164	SR IOV Capabilities			
	0x168	SR IOV Status		SR IOV Control	
	0x16C	TotalVFs (RO)		Initial VF (RO)	
	0x170	Reserved	Function Dependency Link (RO)	Num VF (RW)	
	0x174	VF Stride (RO)		First VF Offset (RO)	
	0x178	VF Device ID		Reserved	
	0x17C	Supported Page Size (0x553)			
	0x180	system page Size (RW)			
	0x184	VF BAR0 - Low (RW)			
	0x188	VF BAR0 - High (RW)			
	0x18C	VF BAR2 (RO)			
	0x190	VF BAR3 - Low (RW)			
	0x194	VF BAR3- High (RW)			
	0x198	VF BAR5 (RO)			
	0x19C	VF Migration State Array Offset (RO)			

An explanation of registers is provided in sections that follow.

9.4 Mandatory PCI Configuration Registers

9.4.1 Vendor ID Register (0x0; RO)

This is a read-only register that has the same value for all PCI functions. It identifies unique Intel products with a value of 0x8086.

9.4.2 Device ID Register (0x2; RO)

This is a read-only register. This field identifies individual 82576 functions. It has the same default value for the two LAN functions but can be auto-loaded from the EEPROM during initialization with different value for each port. The following table describes the possible values according to the SKU and functionality of each function.

For the latest device ID information, see the product specification update.

PCI Function	Default Value	EEPROM Address	Meaning
LAN 0	0x10C9	0x0D	0x10C9 - Dual Port 10/100/1000 Mb/s Ethernet controller, x4 PCIe, copper. 0x10E6 -Dual Port 1000 Mb/s Ethernet controller, x4 PCIe, Fiber. 0x10E7 - Dual Port 10/100/1000 Mb/s Ethernet controller, x4 PCIe, SerDes.
	0x10CA	0x26	0x10CA -Virtual function of 10/100/1000 Mb/s Ethernet controller.
	0x10A6	0x1D	0x10A6 - Dummy function (see note).
LAN 1	0x10C9	0x11	0x10C9 - Dual Port 10/100/1000 Mb/s Ethernet controller, x4 PCIe, copper. 0x10E6 -Dual Port 1000 Mb/s Ethernet controller, x4 PCIe, Fiber. 0x10E7 - Dual Port 10/100/1000 Mb/s Ethernet controller, x4 PCIe, SerDes.
	0x10CA	0x26	0x10CA -Virtual function of 10/100/1000 Mb/s Ethernet controller.
	0x10A6	0x1D	0x10A6 - Dummy function (see note).

Note: The Dummy function device ID is loaded from EEPROM address 0x1D and is used according to the disable status of the function. It is applicable only to function 0. See Section 6.2.6 for details.

9.4.3 Command Register (0x4; R/W)

This is a read/write register. Each function has its own command register. Unless explicitly specified, functionality is the same in all functions.

Bit(s)	R/W	Initial Value	Description
0	R/W	0b	I/O Access Enable For LAN functions this field is R/W. For Dummy function this field is RO as zero.
1	R/W	0b	Memory Access Enable For LAN functions this field is R/W. For Dummy function this field is RO as zero.
2	R/W	0b	Bus Master Enable (BME) For LAN functions this field is R/W. For Dummy function this field is RO as zero.
3	RO	0b	Special Cycle Monitoring Hardwired to 0b.
4	RO	0b	MWI Enable Hardwired to 0b.
5	RO	0b	Palette Snoop Enable Hardwired to 0b.
6	RW	0b	Parity Error Response
7	RO	0b	Wait Cycle Enable Hardwired to 0b.
8	RW	0b	SERR# Enable
9	RO	0b	Fast Back-to-Back Enable Hardwired to 0b.
10	RW	0b	Interrupt Disable[1].
15:11	RO	0x0	Reserved

[1] The Interrupt Disable register bit is a read-write bit that controls the ability of a PCIe function to generate a legacy interrupt message. When set, functions are prevented from generating legacy interrupt messages.

9.4.4 Status Register (0x6; RO)

Each function has its own status register. Unless explicitly specified, functionality is the same in all functions.

Bits	R/W	Initial Value	Description
2:0		000b	Reserved
3	RO	0b	Interrupt Status[1]
4	RO	1b	New Capabilities Indicates that a function implements extended capabilities. The 82576 sets this bit, and implements a capabilities list, to indicate that it supports PCI power management, Message Signaled Interrupts (MSI), Enhanced Message Signaled Interrupts (MSI-X), Vital Product Data (VPD), and the PCIe extensions.
5		0b	66 MHz Capable Hardwired to 0b.
6		0b	Reserved
7		0b	Fast Back-to-Back Capable Hardwired to 0b.
8	R/W1C	0b	Data Parity Reported
10:9		00b	DEVSEL Timing Hardwired to 0b.
11	R/W1C	0b	Signaled Target Abort
12	R/W1C	0b	Received Target Abort
13	R/W1C	0b	Received Master Abort
14	R/W1C	0b	Signaled System Error
15	R/W1C	0b	Detected Parity Error

[1] The *Interrupt Status* field is a RO field that indicates that an interrupt message is pending internally to the function.

9.4.5 Revision Register (0x8; RO)

The default revision ID of the 82576 is 0x01. The value of the rev ID is read from EEPROM word 0x1E. Note that LAN 0 and LAN 1 functions have the same revision ID.

9.4.6 Class Code Register (0x9; RO)

The class code is a RO hard coded value that identifies the 82576's functionality.

- LAN 0, LAN 1 - 0x020000/0x010000 - Ethernet/SCSI Adapter[1]

9.4.7 Cache Line Size Register (0xC; R/W)

This field is implemented by PCIe functions as a read-write field for legacy compatibility purposes but has no impact on any PCIe function functionality. Loaded from EEPROM word 0x1A. All functions are initialized to the same value. In EEPROM-less systems, the value is 0x10.

9.4.8 Latency Timer Register (0xD; RO)

Not used. Hardwired to zero.

[1] Selected according to bit 11 or 12 in word 0x1E in the EEPROM for LAN0 and LAN 1 respectively.

9.4.9 Header Type Register (0xE; RO)

This indicates if a device is single function or multifunction. If a single LAN function is the only active one then this field has a value of 0x00 to indicate a single function device. If other functions are enabled then this field has a value of 0x80 to indicate a multi-function device. The following table lists the different options to set the header type field.

Lan 0 Enabled	Lan 1 Enabled	Cross Mode Enable	Dummy Function Enable	Header Type Expected Value
0	0	X	X	N/A (no function)
1	0	0	X	0x00
0	1	0	0	0x00
0	1	0	1	0x80 (dummy exist)
1	1	X	X	0x80 (dual function)
1	0	1	0	0x00
1	0	1	1	0x80 (dummy exist)
0	1	1	X	0x00

9.4.10 BIST Register (0xF; RO)

BIST is not supported in the 82576.

9.4.11 Base Address Registers (0x10:0x27; R/W)

The Base Address registers (or BARs) are used to map the 82576 register space of the various functions. 32-bit addresses may be used in one register for each memory mapping window or 64-bit addresses with 2 registers for each memory mapping window.

9.4.11.1 32-bit Mapping

This mapping is selected when bits 11:10 in word 0x21 in the EEPROM are equal to 00.

<table>
<tr><th>BAR</th><th>Addr.</th><th>31</th><th>4</th><th>3</th><th colspan="2">2:1</th><th>0</th></tr>
<tr><td>0</td><td>0x10</td><td colspan="2">Memory BAR (R/W - 31:17; 0b - 16:4)</td><td>0</td><td colspan="2">00</td><td>0</td></tr>
<tr><td>1</td><td>0x14</td><td colspan="2">Flash BAR (R/W - 31:23/16; 0b - 22/15:4)
<b>Note:</b> See remark regarding Flash size.</td><td>0</td><td colspan="2">00</td><td>0</td></tr>
<tr><td>2</td><td>0x18</td><td colspan="4">IO BAR (R/W - 31:5; 0b - 4:1)</td><td>0</td><td>1</td></tr>
<tr><td>3</td><td>0x1C</td><td colspan="2">MSI-X BAR (R/W - 31:14; 0b - 13:4)</td><td>0</td><td colspan="2">00</td><td>0</td></tr>
<tr><td>4</td><td>0x20</td><td colspan="6">Reserved (read as all 0b’s)</td></tr>
<tr><td>5</td><td>0x24</td><td colspan="6">Reserved (read as all 0b’s)</td></tr>
</table>

All base address registers have the following fields:

Bit(s)	R/W	Initial Value	Description
0	R	0b for Memory 1b for I/O	0b = Indicates memory space. 1b = Indicates I/O.
2:1	R	00b	Memory Type Indicates the address space size. 00b = 32-bit.
3	R	0b*	Prefetch Memory 0b = Non-prefetchable space 1b = Prefetchable space The 82576 implements non-prefetchable space since it has read-side effects. This bit is read from EEPROM word 0x21 bit 9
31:4	R/W	0x0	Memory Address Space Read/write bits and hardwired to 0b. Depends on the memory mapping window sizes: LAN memory spaces are 128 KB. LAN Flash spaces can be 64 KB, up to 8 MB in powers of two. Mapping window size is set by the EEPROM word 0x0F. MSI-X memory space is 16 KB. IO address space is 32 bytes

9.4.11.2 64-bit Mapping without I/O BAR

This mapping is selected when bits 11:10 in word 0x21 in the EEPROM are equal to 10.

<table>
<tr><th>BAR</th><th>Addr.</th><th>31 4</th><th>3</th><th>2:1</th><th>0</th></tr>
<tr><td>0</td><td>0x10</td><td>Memory BAR (R/W - 31:17; 0b - 16:4)</td><td>0</td><td>00</td><td>0</td></tr>
<tr><td>1</td><td>0x14</td><td colspan="4">Memory BAR high word (R/W - 31:0)</td></tr>
<tr><td>2</td><td>0x18</td><td>Flash BAR (R/W - 31:23/16; 0b - 22/15:4)
<b>Note:</b> See remark regarding Flash size.</td><td>0</td><td>00</td><td>0</td></tr>
<tr><td>3</td><td>0x1C</td><td colspan="4">Flash BAR high word (R/W - 31:0)</td></tr>
<tr><td>4</td><td>0x20</td><td>MSI-X BAR (R/W - 31:14; 0b - 13:4)</td><td>0</td><td>00</td><td>0</td></tr>
<tr><td>5</td><td>0x24</td><td colspan="4">MSI-X BAR high word (R/W - 31:0)</td></tr>
</table>

All even base address registers have the following fields:

Bit(s)	R/W	Initial Value	Description
0	R	0b	0b = Indicates memory space. 1

2:1	R	10b	Memory Type Indicates the address space size. 10b = 64-bit.
3	R	0b*	Prefetch Memory 0b = Non-prefetchable space 1b = Prefetchable space The 82576 implements non-prefetchable space since it has read-side effects. This bit is read from EEPROM word 0x21 bit 9
31:4	R/W	0x0	Memory Address Space Read/write bits and hardwired to 0b. Depends on the memory mapping window sizes: LAN memory spaces are 128 KB. LAN Flash spaces can be 64 KB, up to 8 MB in powers of two. Mapping window size is set by the EEPROM word 0x0F. MSI-X memory space is 16 KB.

All odd base address registers have the following fields:

Bit(s)	R/W	Initial Value	Description
31:0	R/W	0x0	Memory Address Space high bytes.

9.4.11.3 64-bit Mapping Without Flash BAR

This mapping is selected when bits 11:10 of word 0x21 in the EEPROM are equal to 11.

BAR	Addr.	31 4	3	2	1	0
0	0x10	Memory BAR (R/W - 31:17; 0b - 16:4)	0	00		0
1	0x14	Memory BAR high word (R/W - 31:0)				
2	0x18	IO BAR (R/W - 31:5; 0b - 4:1)			0	1
3	0x1C	Reserved				
4	0x20	MSI-X BAR (R/W - 31:14; 0b - 13:4)	0	00		0
5	0x24	MSI-X BAR high word (R/W - 31:0)				

All even base address registers have the following fields:

Bit(s)	R/W	Initial Value	Description
0	R	0b	0b = Indicates memory space. 1b = Indicates I/O space

Bit(s)	R/W	Initial Value	Description
2:1	R	10b - memory 00b- IO	Memory Type Indicates the address space size. 10b = 64-bit for memory BARs 00b = 32-bit for IO BAR
3	R	0b*	Prefetch Memory 0b = Non-prefetchable space 1b = Prefetchable space The 82576 implements non-prefetchable space since it has read-side effects. This bit is read from EEPROM word 0x21 bit 9
31:4	R/W	0x0	Memory Address Space Read/write bits and hardwired to 0b. Depends on the memory mapping window sizes: LAN memory spaces are 128 KB. LAN Flash spaces can be 64 KB, up to 8 MB in powers of two. Mapping window size is set by the EEPROM word 0x0F. MSI-X memory space is 16 KB. IO address space is 32 bytes.

All odd base address registers have the following fields:

Bit(s)	R/W	Initial Value	Description
31:0	R/W	0x0	Memory Address Space high bytes.

Mapping Window	Mapping Description
Memory BAR	The internal registers and memories are accessed as direct memory mapped offsets from the base address register. Software can access Dword or 64 bytes.
Flash BAR	The external Flash can be accessed using direct memory mapped offsets from the Flash base address register. Software can access byte, word, Dword or 64 bytes.
I/OBAR	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: Addr Reg and Data Reg. Software can access byte, word or Dword.
MSI-X BAR	The internal registers and memories are accessed as direct memory mapped offsets from the base address register. Software can access Dword or 64 bytes.

9.4.12 CardBus CIS Register (0x28; RO)

Not used. Hardwired to zero.

9.4.13 Subsystem Vendor ID Register (0x2C; RO)

This value can be loaded automatically from the EEPROM address 0x0C at power up or reset. A value of 0x8086 is the default for this field at power up if the EEPROM does not respond or is not programmed. All functions are initialized to the same value.

9.4.14 Subsystem ID Register (0x2E; RO)

This value can be loaded automatically from EEPROM address 0x0B at power up with a default value of 0x0000.

9.4.15 Expansion ROM Base Address Register (0x30; RO)

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 functions can have this window. It is enabled by the EEPROM words 0x24 and 0x14 for LAN 0 and LAN 1, respectively. This register returns a zero value for functions without an expansion ROM window.

Bit(s)	R/W	Initial Value	Description
0	R/W	0b	Enable 1b = Enables expansion ROM access. 0b = Disables expansion ROM access.
10:1	R	0x0	Reserved Always read as 0b. Writes are ignored.
31:11	R/W	0x0	Address R/W bits and hardwired to 0b. Depends on the memory mapping window size. The LAN expansion ROM spaces can be either 64 KB, up to 8 MB in powers of two. Mapping window size is set by the EEPROM word 0x0F.

9.4.16 Cap_Ptr Register (0x34; RO)

The *Capabilities Pointer* field (Cap_Ptr) is an 8-bit field that provides an offset in the function's PCI configuration space for the location of the first item in the Capabilities Linked List (CLL). The 82576 sets this bit and implements a capabilities list to indicate that it supports PCI power management, Message Signaled Interrupts (MSIs), and PCIe extended capabilities. Its value is 0x40, which is the address of the first entry: PCI power management.

9.4.17 Interrupt Line Register (0x3C; RW)

Read/write register programmed by software to indicate which of the system interrupt request lines this 82576's interrupt pin is bound to. See the PCI definition for more details. Each of the PCI functions has its own register.

9.4.18 Interrupt Pin Register (0x3D; RO)

Read only register.

- LAN 0 / LAN 1 - A value of 0x1 / 0x2 / 0x3 / 0x4 indicates that this function implements legacy interrupt on INTA / INTB / INTC / INTD, respectively. Loaded from EEPROM word 0x24 / 0x14 for LAN 0 and LAN 1, respectively.

Note: If one of the ports is disabled, the remaining port uses INTA, independent of the EEPROM setting.

9.4.19 Max_Lat/Min_Gnt (0x3E; RO)

Not used. Hardwired to zero.

9.5 PCI Capabilities

The first entry of the PCI capabilities link list is pointed by the Cap_Ptr register. The following table describes the capabilities supported by the 82576.

Address	Item	Next Pointer
0x40-47	PCI Power Management	0x50
0x50-67	Message Signaled Interrupt	0x70
0x70-8B	Extended Message Signaled Interrupt	0xA0
0xA0-DB	PCIe Capabilities	0xE0/0x00[1]
0xE0-0xE7	Vital Product Data Capability	0x00

[1] The VPD area in the EEPROM does not exist. In EEPROM-less mode, the PCIe capability is the last one.

9.5.1 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 D3cold state. If aux power is not supplied, the *PME_En* and *PME_Status* fields also reset on exit from D3cold state.

See the detailed description for registers loaded from the EEPROM at initialization time. Some fields in this section depend on the *Power Management Ena* bits in EEPROM word 0x0A.

9.5.1.1 Capability ID Register (0x40; RO)

This field equals 0x01 indicating the linked list item as being the PCI Power Management registers.

9.5.1.2 Next Pointer (0x41; RO)

This field provides an offset to the next capability item in the capability list. Its value of 0x50 points to the MSI capability.

9.5.1.3 Power Management Capabilities - PMC (0x42; RO)

This field describes the 82576's functionality at the power management states as described in the following table.

Bits	R/W	Default	Description
15:11	RO	See value in description column	PME_Support This five-bit field indicates the power states in which the function might assert PME#. Its initial value is loaded from EEPROM word 0x0A Condition Functionality Value PM Dis in EEProm No PME at all states 00000b PM Ena & no Aux Pwr PME at D0 and D3hot 01001b PM Ena w Aux Pwr PME at D0, D3hot and D3cold 11001b
10	RO	0b	D2_Support The 82576 does not support D2 state.
9	RO	0b	D1_Support The 82576 does not support D1 state.
8:6	RO	000b	AUX Current – Required current defined in the Data Register.
5	RO	1b	DSI The 82576 requires its device driver to be executed following transition to the D0 uninitialized state.

Bits	R/W	Default	Description
4	RO	0b	Reserved
3	RO	0b	PME_Clock Disabled. Hardwired to 0b.
2:0	RO	011b	Version The 82576 complies with the PCI PM specification, revision 1.2.

9.5.1.4 Power Management Control / Status Register - PMCSR (0x44; R/W)

This register is used to control and monitor power management events in the 82576.

Bits	R/W	Default	Description
15	R/W1C	0b (at power up)	PME_Status This bit is set to 1b when the function detects a wake-up event independent of the state of the *PME_En* bit. Writing a 1b clears this bit.
14:13	RO	01b	Data_Scale This field indicates the scaling factor to be used when interpreting the value of the Data register. This field equals 01b (indicating 0.1 watt units) if power management is enabled in the EEPROM and the *Data_Select* field is set to 0, 3, 4, 7, (or 8 for Function 0). Otherwise, this field equals 00b.
12:9	R/W	0000b	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 via EEPROM.
8	R/W	0b (at power up)	PME_En If power management is enabled in the EEPROM, writing a 1b to this register enables wake up. If power management is disabled in the EEPROM, writing a 1b to this bit has no affect and does not set the bit to 1b.
7:4	RO	000000b	Reserved
3	RO	0b	No_Soft_Reset This bit is always set to 0b to indicate that the 82576 performs an internal reset after a transition from D3hot to D0 via software control of the *PowerState* bits. Configuration context is lost when performing the soft reset. After transitioning from the D3hot to the D0 state, full re-initialization sequence is needed to return the 82576 to D0 Initialized.
2	RO	0b	Reserved for PCIe.
1:0	R/W	00b	Power State This field is used to set and report the power state of a function as follows: 00b = D0 01b = D1 (cycle ignored if written with the value of 10b) – D2 (cycle ignored if written with this value) 11b = D3 (cycle ignored if power management is not enabled in the EEPROM)

9.5.1.5 Bridge Support Extensions - PMCSR_BSE (0x46; RO)

This register is not implemented in the 82576. Values are set to 0x00.

9.5.1.6 Data Register (0x47; RO)

This optional register is used to report power consumption and heat dissipation. Reported register is controlled by the *Data_Select* field in the PMCSR and the power scale is reported in the *Data_Scale* field in the PMCSR. The data of this field is loaded from the EEPROM if power management is enabled in the EEPROM or with a default value of 0x00. The values for the 82576 functions are read from EEPROM word 0x22.

Function	D0 (Consume/ Dissipate)	D3 (Consume/ Dissipate)	Common
Data Select	0x0 / 0x4	0x3 / 0x7	0x8
Function 0	EEPROM addr 0x22	EEPROM addr 0x22	EEPROM addr 0x22
Function 1	EEPROM addr 0x22	EEPROM addr 0x22	0x00

For other *Data_Select* values, the Data register output is reserved (0b).

9.5.2 MSI Configuration

This structure is required for PCIe functions. There are no changes to this structure.

9.5.2.1 Capability ID Register (0x50; RO)

This field equals 0x05 indicating the linked list item as being the MSI registers.

9.5.2.2 Next Pointer Register (0x51; RO)

This field provides an offset to the next capability item in the capability list. Its value of 0x70 points to the MSI-X capability structure.

9.5.2.3 Message Control Register (0x52; R/W)

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

Bits	R/W	Default	Description
0	R/W	0b	MSI Enable If set to 1b, equals MSI. In this case, the 82576 generates an MSI for interrupt assertion instead of INTx signaling.
3:1	RO	000b	Multiple Message Capable The 82576 indicates a single requested message per each function.
6:4	RO	000b	Multiple Message Enable The 82576 returns 000b to indicate that it supports a single message per function.
7	RO	1b	64-bit capable A value of 1b indicates that the 82576 is capable of generating 64-bit message addresses.
8	RO	1b	MSI per-vector masking. A value of 1b indicates that the 82576 is capable of per-vector masking.
15:9	RO	0b	Reserved Reads as 0b.

9.5.2.4 Message Address Low Register (0x54; 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.

9.5.2.5 Message Address High Register (0x58; R/W)

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

9.5.2.6 Message Data Register (0x5C; 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.

9.5.2.7 Mask Bits Register (0x60; R/W)

The Mask Bits and Pending Bits registers enable software to disable or defer message sending on a per-vector basis. As the 82576 supports only one message, only bit 0 of these register is implemented.

Bits	R/W	Default	Description
0	R/W	0b	MSI Vector 0 Mask If set, the 82576 is prohibited from sending MSI messages.
31:1	RO	000b	Reserved

9.5.2.8 Pending Bits Register (0x64; R/W)

Bits	R/W	Default	Description
0	RO	0b	If set, the 82576 has a pending MSI message.
31:1	RO	000b	Reserved

9.5.3 MSI-X Configuration

More than one MSI-X capability structure per function is prohibited, but a function is permitted to have both an MSI and an MSI-X capability structure.

In contrast to the MSI capability structure, which directly contains all of the control/status information for the function's vectors, the MSI-X capability structure instead points to an MSI-X table structure and a MSI-X Pending Bit Array (PBA) structure, each residing in memory space.

Each structure is mapped by a Base Address Register (BAR) belonging to the function, located beginning at 0x10 in configuration space. A BAR Indicator Register (BIR) indicates which BAR, and a Qword-aligned offset indicates where the structure begins relative to the base address associated with the BAR. The BAR is permitted to be either 32-bit or 64-bit, but must map to memory space. A function is permitted to map both structures with the same BAR, or to map each structure with a different BAR.

The MSI-X table structure, listed in Table 8-20, typically contains multiple entries, each consisting of several fields: message address, message upper address, message data, and vector control. Each entry is capable of specifying a unique vector.

The PBA structure, described in the same section, contains the function's pending bits, one per Table entry, organized as a packed array of bits within Qwords. Note that the last Qword might not be fully populated.

9.5.3.1 Capability ID Register (0x70; RO)

This field equals 0x11 indicating the linked list item as being the MSI-X registers.

9.5.3.2 Next Pointer Register (0x71; RO)

This field provides an offset to the next capability item in the capability list. Its value of 0xA0 points to the PCIe* capability.

9.5.3.3 Message Control Register (0x72; R/W)

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

Bits	R/W	Default	Description
10:0	RO	0x009[1]	TS - Table Size System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of 0x00F indicates a table size of 16.
13:11	RO	0b	Reserved Always returns 000b on read. Write operation has no effect.
14	R/W	0b	FM - Function Mask If set to 1b, all of the vectors associated with the function are masked, regardless of their per-vector *Mask* bit states. If set to 0b, each vector's *Mask* bit determines whether the vector is masked or not. Setting or clearing the *MSI-X Function Mask* bit has no effect on the state of the per-vector *Mask* bits.
15	R/W	0b	En - MSI-X Enable If set to 1b and the *MSI Enable* bit in the MSI Message Control (MMC) register is 0b, the function is permitted to use MSI-X to request service and is prohibited from using its INTx# pin. System configuration software sets this bit to enable MSI-X. A software device driver is prohibited from writing this bit to mask a function's service request. If set to 0b, the function is prohibited from using MSI-X to request service.

[1] Default value is read from the EEPROM

9.5.3.4 Table Offset Register (0x74; R/W)

Bits	R/W	Default	Description
31:3	RO	0x000	Table Offset Used as an offset from the address contained by one of the function's BARs to point to the base of the MSI-X table. The lower three table BIR bits are masked off (set to zero) by software to form a 32-bit Qword-aligned offset.
2:0	RO	0x3	Table BIR Indicates which one of a function's BARs, located beginning at 0x10 in configuration space, is used to map the function's MSI-X table into memory space. A BIR value of 3 indicates that the table is mapped in BAR 3. If 64 bit MMIO mapping is used, this value is set to 4.

9.5.3.5 PBA Offset Register (0x78; R/W)

Bits	R/W	Default	Description
31:3	RO	0x400	PBA Offset Used as an offset from the address contained by one of the function's BARs to point to the base of the MSI-X PBA. The lower three PBA BIR bits are masked off (set to zero) by software to form a 32-bit Qword-aligned offset.
2:0	RO	0x3	PBA BIR: Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X PBA into Memory Space. A BIR value of 3 indicates that the PBA is mapped in BAR 3. If 64 bit MMIO mapping is used, this value is set to 4.

To request service using a given MSI-X table entry, a function performs a Dword memory write transaction using the:

- Contents of the *Message Data* field entry for data.
- Contents of the *Message Upper Address* field for the upper 32 bits of address.
- Contents of the *Message Address* field entry for the lower 32 bits of address.

A memory read transaction from the address targeted by the MSI-X message produces undefined results.

MSI-X table entries and *Pending* bits are each numbered 0 through N-1, where N-1 is indicated by the *Table Size* field in the MMC register. For a given arbitrary MSI-X table entry K, its starting address can be calculated with the formula:

Entry starting address = Table base + K*16

For the associated *Pending* bit K, its address for Qword access and bit number within that Qword can be calculated with the formulas:

Qword address = PBA base + (K div 64)*8
Qword bit# = K mod 64

Software that chooses to read *Pending* bit K with Dword accesses can use these formulas:

Dword address = PBA base + (K div 32)*4
Dword bit# = K mod 32

9.5.4 Vital Product Data Registers

The 82576 supports access to a VPD structure stored in the EEPROM using the following set of registers.

9.5.4.1 Capability ID Register (0xE0; RO)

This field equals 0x3 indicating the linked list item as being the VPD registers.

9.5.4.2 Next Pointer Register (0xE1; RO)

Offset to the next capability item in the capability list. A 0x00 value indicates that it is the last item in the capability-linked list.

9.5.4.3 VPD Address Register (0xE2; RW)

Dword-aligned byte address of the VPD area in the EEPROM to be accessed. The register is read/write with the initial value at power-up indeterminate.

Bits	R/W	Default	Description
14:0	RW	X	Address Dword-aligned byte address of the VPD area in the EEPROM to be accessed. The register is read/write with the initial value at power-up indeterminate. The two LSBs are RO as zero. This is the address relative to the start of the VPD area. As the maximal size supported by the 82576 is 256 bytes, bits 14:8 should always be zero.
15	RW	0b	**F** A flag used to indicate when the transfer of data between the VPD Data register and the storage component completes. The Flag register is written when the VPD Address register is written. 0b = Read. Set by hardware when data is valid. 1b = Write. Cleared by hardware when data is written to the EEPROM. The VPD address and data should not be modified before the action completes.

9.5.4.4 VPD Data Register (0xE4; RW)

This register contains the VPD read/write data.

Bits	R/W	Default	Description
31:0	RW	X	VPD Data VPD data can be read or written through this register. The LSB of this register (at offset four in this capability structure) corresponds to the byte of VPD at the address specified by the VPD Address register. The data read from or written to this register uses the normal PCI byte transfer capabilities. Four bytes are always transferred between this register and the VPD storage component. Reading or writing data outside of the VPD space in the storage component is not allowed. In a write access, the data should be set before the address and the flag is set.

9.5.5 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 82576 implements the PCIe capability structure for endpoint functions as follows:

9.5.5.1 Capability ID Register (0xA0; RO)

This field equals 0x10 indicating the linked list item as being the PCIe Capabilities registers.

9.5.5.2 Next Pointer Register (0xA1; RO)

Offset to the next capability item in the capability list. Its value of 0xE0 points to the VPD structure. If VPD is disabled, a value of 0x00 value indicates that it is the last item in the capability-linked list.

9.5.5.3 PCIe CAP Register (0xA2; RO)

The PCIe capabilities register identifies the PCIe device type and associated capabilities. This is a read only register identical to all functions.

Bits	R/W	Default	Description
3:0	RO	0010b	Capability Version Indicates the PCIe capability structure version number. The 82576 supports both version 1 and version 2 as loaded from the PCIe *Capability Version* bit in the EEPROM.
7:4	RO	0000b	Device/Port Type Indicates the type of PCIe functions. All functions are a native PCI function with a value of 0000b.
8	RO	0b	Slot Implemented The 82576 does not implement slot options therefore this field is hardwired to 0b.
13:9	RO	00000b	Interrupt Message Number The 82576 does not implement multiple MSIs per function, therefore this field is hardwired to 0x0.
15:14	RO	00b	Reserved

9.5.5.4 Device Capability Register (0xA4; RW)

This register identifies the PCIe device specific capabilities. It is a read only register with the same value for the two LAN functions and to all other functions.

Bits	R/W	Default	Description
2:0	RO	010b	Max Payload Size Supported This field indicates the maximum payload that the 82576 can support for TLPs. It is loaded from the EEPROM's PCIe Init Configuration 3 word, 0x1A (with a default value of 512 bytes).
4:3	RO	00b	Phantom Function Supported Not supported by the 82576.
5	RO	0b	Extended Tag Field Supported Max supported size of the *Tag* field. The 82576 supported 5-bit *Tag* field for all functions.
8:6	RO	110b	Endpoint L0s Acceptable Latency This field indicates the acceptable latency that the 82576 can withstand due to the transition from the L0s state to the L0 state. All functions share the same value loaded from the EEPROM PCIe Init Configuration 1 word, 0x18.
11:9	RO	110b	Endpoint L1 Acceptable Latency This field indicates the acceptable latency that the 82576 can withstand due to the transition from the L1 state to the L0 state. All functions share the same value loaded from the EEPROM PCIe Init Configuration 1 word, 0x18.
12	RO	0b	Attention Button Present Hardwired in the 82576 to 0b for all functions.

Bits	R/W	Default	Description
13	RO	0b	Attention Indicator Present Hardwired in the 82576 to 0b for all functions.
14	RO	0b	Power Indicator Present Hardwired in the 82576 to 0b for all functions.
15	RO	1b	Role-Based Error Reporting This bit, when set, indicates that the 82576 implements the functionality originally defined in the Error Reporting ECN for PCIe Base Specification 1.0a and later incorporated into PCIe Base Specification 1.1. Set to 1b in the 82576.
17:16	RO	000b	Reserved
25:18	RO	0x00	Slot Power Limit Value Hardwired in the 82576 to 0x00 for all functions, as the 82576 consumes less than the 25W allowed for it's form factor.
27:26	RO	00b	Slot Power Limit Scale Hardwired in the 82576 to 0b for all functions.as the 82576 consumes less than the 25W allowed for it's form factor.
28	RO	1b	Function Level Reset (FLR) Capability A value of 1b indicates the function supports the optional FLR mechanism.
31:29	RO	000b	Reserved

9.5.5.5 Device Control Register (0xA8; RW)

This register controls the PCIe specific parameters. There is a dedicated register per each function.

Bits	R/W	Default	Description
0	RW	0b	Correctable Error Reporting Enable Enable error report.
1	RW	0b	Non-Fatal Error Reporting Enable Enable error report.
2	RW	0b	Fatal Error Reporting Enable Enable error report.
3	RW	0b	Unsupported Request Reporting Enable Enable error report.
4	RW	1b	Enable Relaxed Ordering If this bit is set, the 82576 is permitted to set the *Relaxed Ordering* bit in the attribute field of write transactions that do not need strong ordering. For more details, refer to the description about the RO_DIS bit in the CTRL_EXT register bit in See Section 8.2.3 .
7:5	RW	000b (128 bytes)	Max Payload Size This field sets maximum TLP payload size for the 82576 functions. As a receiver, the 82576 must handle TLPs as large as the set value. As a transmitter, the 82576 must not generate TLPs exceeding the set value. The max payload size supported in the 82576 capabilities register indicates permissible values that can be programmed.
8	R0	0b	Extended Tag field Enable Not implemented in the 82576.
9	R0	0b	Phantom Functions Enable Not implemented in the 82576.
10	RW	0b	Auxiliary Power PM Enable When set, enables the 82576 to draw AUX power independent of PME AUX power. The 82576 is a multi function device, therefore it is allowed to draw AUX power if at least one of the functions has this bit set.

Bits	R/W	Default	Description
11	RW	1b	Enable No Snoop Snoop is gated by *NONSNOOP* bits in the GCR register in the CSR space.
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 (the default value for non LAN functions). 001b = 256 bytes. 010b = 512 bytes. (the default value for the LAN devices). 011b = 1 KB. 100b = 2 KB. 101b = Reserved. 110b = Reserved. 111b = Reserved.
15	RW	0b	Initiate Function Level Reset A write of 1b initiates an FLR to the function. The value read by software from this bit is always 0b.

9.5.5.6 Device Status Register (0xAA; RW1C)

This register provides information about PCIe device’s specific parameters. There is a dedicated register per each function.

Bits	R/W	Default	Description
0	RW1C	0b	Correctable Detected Indicates status of correctable error detection.
1	RW1C	0b	Non-Fatal Error Detected Indicates status of non-fatal error detection.
2	RW1C	0b	Fatal Error Detected Indicates status of fatal error detection.
3	RW1C	0b	Unsupported Request Detected Indicates that the 82576 received an unsupported request. This field is identical in all functions. The 82576 cannot distinguish which function caused an error.
4	RO	0b	Aux Power Detected If aux power is detected, this field is set to 1b. It is a strapping signal from the periphery identical for all functions. Reset on Internal_Power_On_Reset and PE_RST_N only.
5	RO	0b	Transaction Pending Indicates whether the 82576 has any transaction pending. Transactions include completions for any outstanding non-posted request for all used traffic classes.
15:6	RO	0x00	Reserved

9.5.5.7 Link CAP Register (0xAC; RO)

This register identifies PCIe link specific capabilities. This is a read only register identical to all functions.

Bits	R/W	Default	Description
3:0	RO	0001b	Max Link Speed The 82576 indicates a maximum link speed of 2.5 Gb/s.
9:4	RO	0x4	Max Link Width Indicates the maximum link width. The 82576 can support by 1-, by 2- and by 4-link width. The field is loaded from the EEPROM PCIe Init Configuration 3, word 1Ah, with a default value of four lanes. Relevant encoding: 000000b = Reserved. 000001b = x1. 000010b = x2. 000100b = x4.
11:10	RO	11b	Active State Link PM Support Indicates the level of active state power management supported in the 82576. The encoding is: 00b = Reserved. 01b = L0s Entry Supported. 10b = Reserved. 11b = L0s and L1 Supported. This field is loaded from the EEPROM PCIe Init Configuration 3 word, 0x1A.
14:12	RO	101b (1 µs – 2 µs) when non common clock 110b (2 µs – 4 µs) when common clock	L0s Exit Latency Indicates the exit latency from L0s to L0 state. 000b = Less than 64ns. 001b = 64ns – 128ns. 010b = 128ns – 256ns. 011b = 256ns - 512ns. 100b = 512ns - 1 µs. 101b = 1 µs – 2 µs. 110b = 2 µs – 4 µs. 111b = Reserved. If the 82576 uses a common clock the value of this field is loaded from PCIe Init Config 1 word, 0x18, bits [2:0]. If the 82576 uses a separate clock, the value of this field is loaded from PCIe Init Config 1 word, 0x18, bits [5:3].
17:15	RO	110b (32-64 µs)	L1 Exit Latency Indicates the exit latency from L1 to L0 state. This field is loaded from the EEPROM PCIe Init Configuration 1 word, 0x18. 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.
18	RO	0b	Clock Power Management Status Not supported in the 82576. RO as zero.
19	RO	0b	Surprise Down Error Reporting Capable Status Not supported in the 82576. RO as zero
20	RO	0b	Data Link Layer Link Active Reporting Capable Status Not supported in the 82576. RO as zero.

Bits	R/W	Default	Description
21	RO	0b	Link Bandwidth Notification Capability Status Not supported in the 82576. RO as zero.
23:22	RO	00b	Reserved
31:24	HwInit	0x0	Port Number The PCIe port number for the given PCIe link. Field is set in the link training phase.

9.5.5.8 Link Control Register (0xB0; RO)

This register controls PCIe link specific parameters. There is a dedicated register per each function.

Bits	R/W	Default	Description
1:0	RW	00b	Active State Link PM Control This field controls the active state of power management that is supported on the link. Link PM functionality is determined by the lowest common denominator of all functions. The encoding is: 00b = PM disabled. 01b = L0s entry supported. 10b = Reserved. 11b = L0s and L1 supported.
2	RO	0b	Reserved
3	RW	0b	Read Completion Boundary
4	RO	0b	Link Disable Not applicable for endpoint devices; hardwired to 0b.
5	RO	0b	Retrain Clock Not applicable for endpoint devices; hardwired to 0b.
6	RW	0b	Common Clock Configuration When this bit is set, it indicates that the 82576 and the component at the other end of the link are operating with a common reference clock. A value of 0b indicates that both operate with an asynchronous clock. This parameter affects the L0s exit latencies.
7	RW	0b	Extended Synch When this bit is set, it forces an extended Tx of a FTS ordered set in FTS and an extra TS1 at exit from L0s prior to enter L0.
8	RO	0b	Enable Clock Power Management Not supported in the 82576. RO as zero.
9	RO	0b	Hardware Autonomous Width Disable Not supported in the 82576. RO as zero.
10	RO	0b	Link Bandwidth Management Interrupt Enable Not supported in the 82576. RO as zero.
11	RO	0b	Link Autonomous Bandwidth Interrupt Enable Not supported in the 82576. RO as zero.
15:12	RO	0000b	Reserved

9.5.5.9 Link Status Register (0xB2; RO)

This register provides information about PCIe link specific parameters. This is a read only register identical to all functions.

Bits	R/W	Default	Description
3:0	RO	0001b	Link Speed Indicates the negotiated link speed. Note that the default setting (0001b) is the only defined speed which is 2.5 Gb/s.
9:4	RO	000001b	Negotiated Link Width Indicates the negotiated width of the link. Relevant encoding for the 82576 are: 000001b = x1 000010b = x2 000100b = x4
10	RO	0b	Reserved (was: Link Training Error)
11	RO	0b	Link Training Indicates that link training is in progress.
12	HwInit	1b	Slot Clock Configuration When set, indicates that the 82576 uses the physical reference clock that the platform provides on the connector. This bit must be cleared if the 82576 uses an independent clock. The Slot Clock Configuration bit is loaded from the *Slot_Clock_Cfg* EEPROM bit.
13	RO	0b	Data Link Layer Link Active Not supported in the 82576. RO as zero.
14	RO	0b	Link Bandwidth Management Status Not supported in the 82576. RO as zero.
15	RO	0b	Reserved

9.5.5.10 Reserved Registers (0xB4-0xC0; RO)

Unimplemented reserved registers not relevant to PCIe endpoint.

The two following registers are implemented only if the capability version is 2.

9.5.5.11 Device CAP 2 Register (0xC4; RO)

This register identifies PCIe device specific capabilities. It is a read only register with the same value for all functions.

Bit Location	R/W	Default	Description
3:0	RO	1111b	Completion Timeout Ranges Supported This field indicates 82576 support for the optional completion timeout programmability mechanism. This mechanism enables system software to modify the completion timeout value. Four time value ranges are defined: Range A = 50 µs to 10 ms Range B = 10 ms to 250 ms Range C = 250 ms to 4 s Range D = 4 s to 64 s Bits are set according to the table in Section 9.5.5.12 to show the timeout value ranges that are supported. 0000b = Completion timeout programming not supported. the 82576 must implement a timeout value in the range 50 µs to 50 ms. 0001b = Range A. 0010b = Range B. 0011b = Ranges A & B. 0110b = Ranges B & C. 0111b = Ranges A, B & C. 1110b = Ranges B, C & D. 1111b = Ranges A, B, C & D. All other values are reserved. It is strongly recommended that the completion timeout mechanism not expire in less than 10 ms
4	RO	1b	Completion Timeout Disable Supported A value of 1b indicates support for the completion timeout disable mechanism.
5	RO	0b	ARI Forwarding Supported Applicable only to switch downstream ports and root ports; must be set to 0b for other function types.
15:5	RO	0x0	Reserved. Set to 0x0.

9.5.5.12 Device Control 2 Register (0xC8; RW)

This register controls PCIe specific parameters. There is a dedicated register per each function.

Bit location	R/W	Default	Description
3:0	RW	0000b	Completion Timeout Value In devices that support completion timeout programmability, this field enables system software to modify the completion timeout value. Encoding: 0000b = Default range: 50 µs to 50 ms. It is strongly recommended that the completion timeout mechanism not expire in less than 10 ms. Values available if Range A (50 µs to 10 ms) programmability range is supported: 0001b = 50 µs to 100 µs. 0010b = 1 ms to 10 ms. Values available if Range B (10 ms to 250 ms) programmability range is supported: 0101b = 16 ms to 55 ms. 0110b = 65 ms to 210 ms. Values available if Range C (250 ms to 4 s) programmability range is supported: 1001b = 260 ms to 900 ms. 1010b = 1 s to 3.5 s. Values available if the Range D (4 s to 64 s) programmability range is supported: 1101b = 4 s to 13 s. 1110b = 17 s to 64 s. Values not defined are reserved. Software is permitted to change the value in this field at any time. For requests already pending when the completion timeout value is changed, hardware is permitted to use either the new or the old value for the outstanding requests and is permitted to base the start time for each request either when this value was changed or when each request was issued. The default value for this field is 0000b.
4	RW	0b	Completion Timeout Disable When set to 1b, this bit disables the completion timeout mechanism. Software is permitted to set or clear this bit at any time. When set, the completion timeout detection mechanism is disabled. If there are outstanding requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding requests. If this is done, it is permitted to base the start time for each request on either the time this bit was cleared or the time each request was issued. The default value for this bit is 0b.
5	RO	0b	Alternative RID Interpretation (ARI) Forwarding Enable Applicable only to switch devices.
15:5	RO	0x0	Reserved

9.6 PCIe Extended Configuration Space

PCIe extended 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 82576 decodes an additional 4-bits (bits 27:24) to provide the additional configuration space as shown in Table 9-6. 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 a PCI-compatible bus, device, and function numbers as follows.

Table 9-6. PCIe Extended Configuration Space

31 28	27 20	19 15	14 12	11 2	1 0
0000b	Bus #	Device #	Fun #	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 0x100 in the function configuration space. The first Dword of the capability structure identifies the capability/version and points to the next capability.

The 82576 supports the following PCIe extended capabilities.

Table 9-7. PCIe Extended Capability Structure

Capability	Offset	Next Header
Advanced Error Reporting Capability	0x100	0x140/0x150/0x000[1]
Serial Number[2]	0x140	0x150/0x000[1]
Alternative RID Interpretation (ARI)	0x150	0x160
IOV support	0x160	0x000

1. Depends on EEPROM settings enabling the serial numbers and ARI/IOV structures.
2. Not available in EEPROM-less systems.

9.6.1 Advanced Error Reporting (AER) Capability

The PCIe AER capability is an optional extended capability to support advanced error reporting. The following table lists the PCIe AER extended capability structure for PCIe functions.

Table 9-8. PCIe AER extended Capability Structure

Register Offset	Field	Description
0x100	PCIe CAP ID	PCIe Extended Capability ID
0x104	Uncorrectable Error Status	Reports error status of individual uncorrectable error sources on a PCIe device.
0x108	Uncorrectable Error Mask	Controls reporting of individual uncorrectable errors by device to the host bridge via a PCIe error message.
0x10C	Uncorrectable Error Severity	Controls whether an individual uncorrectable error is reported as a fatal error.
0x110	Correctable Error Status	Reports error status of individual correctable error sources on a PCIe device.
0x114	Correctable Error Mask	Controls reporting of individual correctable errors by device to the host bridge via a PCIe error message.
0x118	Advanced Error Capabilities and Control Register	Identifies the bit position of the first uncorrectable error reported in the Uncorrectable Error Status register.
0x11C: 0x128	Header Log	Captures the header for the transaction that generated an error.

9.6.1.1 PCIe CAP ID Register (0x100; RO)

Bit Location	R/W	Default Value	Description
15:0	RO	0x0001	Extended Capability ID PCIe extended capability ID indicating AER capability.
19:16	RO	0x1	Version Number PCIe AER extended capability version number.
31:20	RO	0x150	Next Capability Pointer Next PCIe extended capability pointer. A value of 0x140 points to the serial ID capability. In EEPROM-less systems or when serial ID is disabled in the EEPROM, the next pointer is 0x150 and points to the ARI capability structure. If ARI/IOV and serial ID are disabled in the EEPROM this field is 0x0.

9.6.1.2 Uncorrectable Error Status Register (0x104; R/W1CS)

The Uncorrectable Error Status register reports error status of individual uncorrectable error sources on a PCIe function. An individual error status bit that is set to 1b indicates that a particular error occurred; software can clear an error status by writing a 1b to the respective bit.

Bit Location	R/W	Default Value	Description
3:0	RO	0x0	Reserved
4	R/W1CS	0b	Data Link Protocol Error Status
5	RO	0b	Surprise Down Error Status (Optional) Not supported in the 82576.
11:6	RO	0x0	Reserved
12	R/W1CS	0b	Poisoned TLP Status
13	R/W1CS	0b	Flow Control Protocol Error Status
14	R/W1CS	0b	Completion Timeout Status
15	R/W1CS	0b	Completer Abort Status
16	R/W1CS	0b	Unexpected Completion Status
17	R/W1CS	0b	Receiver Overflow Status
18	R/W1CS	0b	Malformed TLP Status
19	RO	0b	ECRC Error Status Not supported in the 82576.
20	R/W1CS	0b	Unsupported Request Error Status When caused by a function that claims a TLP
21	RO	0b	ACS Violation Status Not supported in the 82576.
31:22	RO	0x0	Reserved

9.6.1.3 Uncorrectable Error Mask Register (0x108; RWS)

The Uncorrectable Error Mask register controls reporting of individual uncorrectable errors by device to the host bridge via a PCIe error message. A masked error (respective bit set in mask register) is not reported to the host bridge by an individual device. There is a mask bit per bit in the Uncorrectable Error Status register.

Bit Location	R/W	Default Value	Description
3:0	RO	0x0	Reserved
4	RWS	0b	Data Link Protocol Error Mask
11:5	RO	0x0	Reserved
12	RWS	0b	Poisoned TLP Mask
13	RWS	0b	Flow Control Protocol Error Mask
14	RWS	0b	Completion Timeout Mask
15	RWS	0b	Completer Abort Mask
16	RWS	0b	Unexpected Completion Mask
17	RWS	0b	Receiver Overflow Mask
18	RWS	0b	Malformed TLP Mask
19	RO	0b	Reserved
20	RWS	0b	Unsupported Request Error Mask
31:21	RO	0x0	Reserved

9.6.1.4 Uncorrectable Error Severity Register (0x10C; RWS)

The Uncorrectable Error Severity register controls whether an individual uncorrectable error is reported as a fatal error. An uncorrectable error is reported as fatal when the corresponding error bit in the severity register is set. If the bit is cleared, the corresponding error is considered non-fatal.

Bit Location	R/W	Default Value	Description
3:0	RO	0001b	Reserved
4	RWS	1b	Data Link Protocol Error Severity
11:5	RO	0x0	Reserved
12	RWS	0b	Poisoned TLP Severity
13	RWS	1b	Flow Control Protocol Error Severity
14	RWS	0b	Completion Timeout Severity
15	RWS	0b	Completer Abort Severity
16	RWS	0b	Unexpected Completion Severity
17	RWS	1b	Receiver Overflow Severity
18	RWS	1b	Malformed TLP Severity
19	RO	0b	Reserved
20	RWS	0b	Unsupported Request Error Severity
31:21	RO	0x0	Reserved

9.6.1.5 Correctable Error Status Register (0x110; R/W1CS)

The Correctable Error Status register reports error status of individual correctable error sources on a PCIe device. When an individual error status bit is set to 1b, it indicates that a particular error occurred; software can clear an error status by writing a 1b to the respective bit.

Bit Location	R/W	Default Value	Description
0	R/W1CS	0b	Receiver Error Status
5:1	RO	0x0	Reserved
6	R/W1CS	0b	Bad TLP Status
7	R/W1CS	0b	Bad DLLP Status
8	R/W1CS	0b	REPLAY_NUM Rollover Status
11:9	RO	000	Reserved
12	R/W1CS	0b	Replay Timer Timeout Status
13	R/W1CS	0b	Advisory Non-Fatal Error Status
31:14	RO	0x0	Reserved

9.6.1.6 Correctable Error Mask Register (0x114; RWS)

The Correctable Error Mask register controls reporting of individual correctable errors by device to the host bridge via a PCIe error message. A masked error (respective bit set in mask register) is not reported to the host bridge by an individual device. There is a mask bit per bit in the Correctable Error Status register.

Bit Location	R/W	Default Value	Description
0	RWS	0b	Receiver Error Mask
5:1	RO	0x0	Reserved
6	RWS	0b	Bad TLP Mask
7	RWS	0b	Bad DLLP Mask
8	RWS	0b	REPLAY_NUM Rollover Mask
11:9	RO	000b	Reserved
12	RWS	0b	Replay Timer Timeout Mask
13	RWS	0b	Advisory Non-Fatal Error Mask
31:14	RO	0x0	Reserved

9.6.1.7 Advanced Error Capabilities and Control Register (0x118; RO)

Bit Location	R/W	Default Value	Description
4:0	RO	0x0	Vector pointing to the first recorded error in the Uncorrectable Error Status register.
5	RO	0b	ECRC Generation Capable This bit indicates that the 82576 is capable of generating ECRC. Tied to 0b in the 82576.
6	RO	0b	ECRC Generation Enable This bit, when set, enables ECRC generation. Tied to 0b in the 82576.
7	RO	0b	ECRC Check Capable This bit indicates that the 82576 is capable of checking ECRC. Tied to 0b in the 82576.
8	RO	0b	ECRC Check Enable This bit, when set, enables ECRC checking. Tied to 0b in the 82576.
31:9	RO	0x0	Reserved

9.6.1.8 Header Log Register (0x11C:0x128; RO)

The Header Log register captures the header for the transaction that generated an error. This register is 16 bytes in length.

Bit Location	R/W	Default Value	Description
127:0	RO	0b	Header of the packet in error (TLP or DLLP).

9.6.2 Serial Number

The PCIe device serial number capability is an optional extended capability implemented by the 82576. The device serial number is a read-only 64-bit value that is unique for a given PCIe device.

Both functions return the same device serial number value.

9.6.2.1 Device Serial Number Enhanced Capability Header Register (0x140; RO)

The following table lists the allocation of register fields in the device serial number enhanced capability header. It also lists the respective bit definitions. The extended capability ID for the device serial number capability is 0x0003.

9.6.2.2 Serial Number Register (0x144:0x148; RO)

The Serial Number register is a 64-bit field that contains the IEEE defined 64-bit extended unique identifier (EUI-64). Table 9-9 lists the allocation of register fields in the Serial Number register. Table 9-9 also lists the respective bit definitions.

Bit(s) Location	Default value	R/W	Description
15:0	0x0003	RO	PCIe Extended Capability ID This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability. Extended capability ID for the device serial number.
19:16	0x1	RO	Capability Version This field is a PCI-SIG defined version number that indicates the version of the current capability structure.
31:20	0x150	RO	Next Capability Offset This field contains the offset to the next PCIe capability structure or 0x000 if no other items exist in the linked list of capabilities. The value of this field is 0x150 to point to the ARI capability structure. If ARI/IOV and serial ID are disabled in the EEPROM, then this field is 0x0.

Table 9-9. Serial Number Register

31:0

Serial Number Register (Lower Dword)
Serial Number Register (Upper word)

63:32

Serial number definition in the 82576:

Table 9-10. SN Definition

Bit(s) Location	R/W	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 ID value assigned by IEEE registration authority and a 40-bit extension identifier assigned by the manufacturer.

Serial number uses the MAC address according to the following definition:

Table 9-11. SN and MAC Address

Field	Extension identifier					Company ID		
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:

Table 9-12. SN Constructed from 48-bit MAC Address

Field	Extension identifier			MAC Label		Company ID		
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 0xFFFF.

For example, assume that the company ID is (Intel) 00-A0-C9 and the extension identifier is 23-45-67. In this case, the 64-bit serial number is:

Table 9-13. Example 64-bit SN

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

Most significant byte Least significant byte

Most significant bit Least significant bit

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

The translation from EEPROM words 0 to 2 to the serial number is as follows:

- Serial number ADDR+0 = EEPROM byte 5
- Serial number ADDR+1 = EEPROM byte 4
- Serial number ADDR+2 = EEPROM byte 3
- Serial number ADDR+3,4 = 0xFF 0xFF
- Serial number ADDR+5 = EEPROM byte 2
- Serial number ADDR+6 = EEPROM byte 1
- Serial number ADDR+7 = EEPROM byte 0

The official document defining EUI-64 is: http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

9.6.3 ARI Capability Structure

In order to enable more than eight functions per end point without requesting an internal switch (typically needed in virtualization scenarios), the PCI sig defines a new capability that enables a different interpretation of the *Bus*, *Device*, and *Function* fields. The Alternate Requester ID Interpretation (ARI) capability structure is as follows:

9.6.3.1 PCIe ARI Header Register (0x150; RO)

Bit(s)	Initial Value	R/W	Description
15:0	0x000E	RO	ID - PCIe Extended Capability ID PCIe extended capability ID for the ARI.
19:16	0x1	RO	Version - Capability Version This field is a PCI-SIG defined version number that indicates the version of the current capability structure. Must be 0x1 for this version of the specification.
31:20	0x160	RO	Next Capability Ptr. - Next Capability Offset This field contains the offset to the next PCIe extended capability structure. The value of the 0x160 points to the IOV structure.

9.6.3.2 PCIe ARI Capabilities & Control Register (0x154; RO)

Bit(s)	R/W	Initial Value	Description
0	RO	0b	M - MFVC Function Groups Capability Applicable only to function 0; must be 0b for all other functions. If 1b, indicates that the ARI device supports function group level arbitration via its Multi-Function Virtual Channel (MFVC) Capability structure. Not supported in the 82576.
1	RO	0b	A - ACS Function Groups Capability (A) Applicable only to function 0; must be 0b for all other functions. If 1b, indicates that the ARI device supports function group level granularity for ACS P2P egress control via its ACS capability structures. Not supported in the 82576.
7:2	RO	0x0	Reserved
15:8	RO	0x1 (func 0) 0x0 (func 1)[1]	NFP - Next Function Pointer This field contains the pointer to the next physical function configuration space or 0x0000 if no other items exist in the linked list of functions. Function 0 is the start of the link list of functions.
16	RO	0b	M_EN - MFVC Function Groups Enable (M) Applicable only for function 0; must be hardwired to 0b for all other functions. When set, the ARI device must interpret entries in its function arbitration table as function group numbers rather than function numbers. Not supported in the 82576.
17	RO	0b	A_EN - ACS Function Groups Enable (A) Applicable only for function 0; must be hardwired to 0b for all other functions. When set, each function in the ARI device must associate bits within its egress control vector with function group numbers rather than function numbers. Not supported in the 82576.
19:18	RO	00b	Reserved
22:20	RO	0x0	Function Group Number (FGN) Not supported in the 82576.
31:23	RO	0x0	Reserved

[1] If port 0 and port 1 are switched or function zero is a dummy function, this register should keep it's attributes according to the function number. If LAN1 is disabled, then the value of this field in function zero should be zero.

9.6.4 IOV Capability Structure

This is the new structure used to support the IOV capabilities reporting and control.

9.6.4.1 PCIe SR-IOV Header Register (0x160; RO)

Bit(s)	R/W	Initial Value	Description
15:0	RO	0x0010	PCIe Extended Capability ID PCIe extended capability ID of the SR-IOV capability.
19:16	RO	0x1	Capability Version This field is a PCI-SIG defined version number that indicates the version of the current capability structure. Must be 0x1 for this version of the specification.
31:20	RO	0x0	Next Capability Offset This field contains the offset to the next PCIe extended capability structure or 0x000 if no other items that exist in the linked list of capabilities.

9.6.4.2 PCIe SR-IOV Capabilities Register (0x164; RO)

Bit(s)	R/W	Initial Value	Description
0	RO	0b	VF Migration Capable Migration capable device running under migration capable MR-PCIM. RO as 0b in the 82576.
20:1	RO	0x0	Reserved
31:21	RO	0x0	VF Migration Interrupt Message Number Indicates the MSI/MSI-X vector used for the interrupts. This field is undefined when the VF Migration Capable bit is cleared.

9.6.4.3 PCIe SR-IOV Control Register (0x168; RW)

Bit(s)	R/W	Initial Value	Description
0	RW	0b	VFE: VF Enable/Disable VF Enable manages the assignment of VFs to the associated PF. If VF Enable is Set, the VFs associated with the PF are accessible in the PCIe fabric. When Set, VFs respond to and may issue PCI-Express transactions following the rules for PCI-Express Endpoint Functions. If Clear, VFs are disabled and not visible in the PCI-Express fabric; VFs shall respond to Requests with UR and may not issue PCIe transactions. Setting VF Enable after it has previously been Cleared shall result in the same VF state as if FLR had been issued to the VF.
1	RO	0b	VF ME - VF Migration Enable Enables/disables VF migration support.
2	RO	0b	VF MIE - VF Migration Interrupt Enable Enables/disables VF migration state change interrupt.
3	RW	0b	VF MSE - Memory Space Enable for Virtual Functions VF MSE controls memory space enable for all VFs associated with this PF as with the *Memory Space Enable* bit in a functions PCI command register. The default value for this bit is 0b. When *VF Enable* = 1b, virtual function memory space access is permitted only when VF MSE is set. VFs must follow the same error reporting rules as defined in the base specification if an attempt is made to access a virtual functions memory space when *VF Enable* is 1b and VF MSE is 0b. **Implementation Note:** Virtual functions memory space cannot be accessed when the *VF Enable* bit = 0b. Thus, VF MSE is a don't care when *VF Enable* = 0b, however, software might choose to set VF MSE after programming the VF BARn registers, prior to setting *VF Enable* to 1b.

Bit(s)	R/W	Initial Value	Description
4	RW (func 0) RO (func 1)[1]	0b	VF ARI - VF ARI Enable The 82576 can locate VFs in function numbers 8 to 255 of the captured bus number. This bit is not reset by an FLR reset - only by a full device reset.
15:5	RO	0x0	Reserved
16	RO	0b	VFMIS - VF Migration Event Pending Indicates a VF migration in or migration out request has been issued by MR-PCIM. To determine the cause of the event, software can scan the VF state array. Not implemented in the 82576.
31:17	RO	0x0	Reserved

[1] If the ports are switched, this field should keep it's attributes according to the function number.

9.6.4.4 PCIe SR-IOV Max/Total VFs Register (0x16C)

Bit(s)	R/W	Initial Value	Description
15:0	RO	0x8	**InitialVFs** **InitialVFs** indicates the number of VFs that are initially associated with the PF. If VF migration capable is clear, this field must contain the same value as TotalVFs. A lower value of this field can be loaded from the IOV control word in the EEPROM.
31:16	RO	0x8	**TotalVFs** **TotalVFs** indicates the maximum number of VFs that could be associated with the PF. In the 82576, this is equal to InitialVFs.

9.6.4.5 PCIe SR-IOV Num VFs Register (0x170; R/W)

Bit(s)	R/W	Initial Value	Description
15:0	R/W	0x0	**NumVFs** **NumVFs** define the number of VFs software has assigned to the PF. Software sets NumVFs as part of the process of creating VFs. NumVFs VFs must be visible in the PCIe fabric after both NumVFs are set to a valid value and *VF Enable* is set to 1b. Visible in the PCIe fabric means that the VF must respond to PCIe transactions targeting the VF, following all other rules defined by this specification and the base specification. The results are undefined if NumVFs are set to a value greater than TotalVFs. NumVFs can only be written while *VF Enable* is clear. The NumVFs field is RO when *VF Enable* is set.
23:16	RO	0x0 (func 0) 0x1 (func 1)[1]	**FDL - Function Dependency Link** Defines dependencies between physical functions allocation. The default behavior of the 82576 is not to define any such constraints.
31:24	RO	0x0	Reserved

[1] Even if port 0 and port 1 are switched or function zero is a dummy function, this register should keep it's attributes according to the function number.

9.6.4.6 PCIe SR-IOV VF RID Mapping Register (0x174; RO)

See Section 7.9.2.6 for details of the RID mapping.

Bit(s)	R/W	Initial Value	Description
15:0	RO	0x180	**FVO** **First VF offset** defines the requestor ID (RID) offset of the first VF that is associated with the PF that contains this capability structure. The first VFs 16-bit RID is calculated by adding the contents of this field to the RID of the PF containing this field. The content of this field is valid only when *VF Enable* is set. If *VF Enable* is 0b, the contents are undefined. If the *ARI Enable* bit is set, this field changes to 0x80.
31:16	RO	0x2	**VFS** **VF Stride** defines the Requestor ID (RID) offset from one VF to the next one for all VFs associated with the PF that contains this capability structure. The next VFs 16-bit RID is calculated by adding the contents of this field to the RID of the current VF. The content of this field is valid only when *VF Enable* is set and NumVFs are a non-zero. If *VF Enable* is 0b or if NumVFs are zero, the contents are undefined.

9.6.4.7 PCIe SR-IOV VF Device ID Register (0x178; RO)

Bit(s)	R/W	Initial Value	Description
31:16	RO	0x10CA	This field contain the Device ID that should be presented for every VF to the Virtual Machine software. The value of this field may be read from EEPROM word 0x26
15:0	RO	0	Reserved

9.6.4.8 PCIe SR-IOV Supported Page Size Register (0x17C; RO)

Bit(s)	R/W	Initial Value	Description
31:0	RO	0x553	Supported page Size For PFs that support the stride-based BAR mechanism, this field defines the supported page sizes. This PF supports a page size of 2^(n+12) if bit n is set. For example, if bit 0 is set, the EP supports 4 KB page sizes. Endpoints are required to support 4 KB, 8 KB, 64 KB, 256 KB, 1 MB and 4 MB page sizes. All other page sizes are optional.

9.6.4.9 PCIe SR-IOV System Page Size Register (0x180; R/W)

Bit(s)	R/W	Initial Value	Description
31:0	R/W	0x1	Page Size This field defines the page size the system uses to map the PF's and associated VFs' memory addresses. Software must set the value of the system page size to one of the page sizes set in the *Supported Page Sizes* field. As with supported page sizes, if bit n is set in system page size, the PF and its associated VFs are required to support a page size of 2^(n+12). For example, if bit 1 is set, the system is using an 8 KB page size. The results are undefined if more than one bit is set in system page size. The results are undefined if a bit is set in a system page size that is not set in supported page sizes. When system page size is set, the PF and associated VFs are required to align all BAR resources on a system page size boundary. Each BAR size, including VF BARn size (described in the sections that follow) must be aligned on a system page size boundary. Each BAR size, including VF BARn size must be sized to consume a multiple of system page size bytes. All fields requiring page size alignment within a function must be aligned on a system page size boundary. *VF Enable* must be set to 0b when system page size is set. The results are undefined if system page size is set when *VF Enable* is set.

9.6.4.10 PCIe SR-IOV BAR 0 - Low Register (0x184; R/W)

Bit(s)	R/W	Initial Value	Description
0	RO	0b	Mem 0b = Indicates memory space.
2:1	RO	10b	Mem Type Indicates the address space size. 10b = 64-bit. BAR bit sizes are set according to bit 2 in EEPROM word 0x25.
3	RO	0b	Prefetch Mem 0b = Non-prefetchable space. 1b = Prefetchable space. This BARs prefetchable bit is set according to bit 1 in EEPROM word 0x25.
31:4	R/W	0x0	Memory Address Space Which bits are R/W bits and which are read only to 0b depends on the memory mapping window size. The size is a maximum between 16 KB and the page size.

9.6.4.11 PCIe SR-IOV BAR 0 - High Register (0x188; R/W)

Bit(s)	R/W	Initial Value	Description
31:0	RW	0b	BAR0 - MSB MSB part of BAR 0.

9.6.4.12 PCIe SR-IOV BAR 2 Register (0x18C; RO)

Bit(s)	R/W	Initial Value	Description
31:0	RO	0b	BAR2 This BAR is not used.

9.6.4.13 PCIe SR-IOV BAR 3 - Low Register (0x190; R/W)

Bit(s)	R/W	Initial Value	Description
0	RO	0b	Mem 0b = Indicates memory space.
2:1	RO	10b	Mem Type Indicates the address space size. 10b = 64-bit. BAR bit sizes are set according to bit 2 in EEPROM word 0x25.
3	RO	0b	Prefetch Mem 0b = Non-prefetchable space. 1b = Prefetchable space. This BAR's prefetchable bit is set according to bit 1 in EEPROM word 0x25.
31:4	R/W	0b	Memory Address Space Which bits are R/W bits and which are read only to 0b depends on the memory mapping window size. The size is a maximum between 16 KB and the page size.

9.6.4.14 PCIe SR-IOV BAR 3 - High Register (0x194; R/W)

Bit(s)	R/W	Initial Value	Description
31:0	RW	0x0	BAR3 - MSB MSB part of BAR 3.

9.6.4.15 PCIe SR-IOV BAR 5 Register (0x198; RO)

Bit(s)	R/W	Initial Value	Description
31:0	RO	0x0	BAR5 This BAR is not used.

9.6.4.16 PCIe SR-IOV VF Migration State Array Offset Register (0x19C; RO)

Bit(s)	R/W	Initial Value	Description
2:0	RO	000b	BIR: Indicates which PF BAR contains the VF migration state array. Not implemented in the 82576.
31:0	RO	0x0	Offset, relative to the beginning of the BAR of the start of the migration array. Not implemented in the 82576.

9.7 Virtual Functions (VF) Configuration Space

The configuration space reflected to each VF is a sparse version of the physical function configuration space. Table 9-14 lists the behavior of each register in the VF configuration space.

Table 9-14. VF PCIe Configuration Space

Section	Offset	Name	VF Behavior	Notes
PCI Mandatory Registers	0	Vendor ID	RO - 0xFFFF	
	2	Device ID	RO - 0xFFFF	
	4	Command	RW	See Section 9.7.1.1 for details
	6	Status	Per VF	See Section 9.7.1.2 for details
	8	RevisionID	RO as PF	
	9	Class Code	RO as PF	
	C	Cache Line Size	RO - 0	
	D	LatencyTimer	RO - 0	
	E	Header Type	RO - 0	
	F	BIST	RO - 0	
	10 - 27	BARs	RO - 0	Emulated by VMM
	28	CardBus CIS	RO - 0	Not used
	2C	Sub Vendor ID	RO as PF	
	2E	Sub System	RO as PF	
	30	Expansion ROM	RO - 0	Emulated by VMM
	34	Cap Pointer	RO - 0x70	Points to MSI-X
	3C	Int Line	RO - 0	
	3D	Int Pin	RO - 0	
	3E	Max Lat/Min Gnt	RO - 0	
MSI-X Capability	70	MSI-X Header	RO - 0xA011	Points to PCIe capability
	72	MSI-x Message Control	per VF	See Section 9.7.2.1.1
	74	MSI-X Table Address	RO - as PF	
	78	MSI-X PBA Address	RO	
PCIe Capability	A0	PCIe Header	RO - 0x0010	Last capability
	A2	PCIe Capabilities	RO - as PF	0x0002
	A4	PCIe Dev Cap	RO - as PF	
	A8	PCIe Dev Ctrl	RW	RO as zero apart from FLR - see Section 9.7.2.2.1
	AA	PCIe Dev Status	per VF	See Section 9.7.2.2.2
	AC	PCIe Link Cap	RO - as PF	
	B0	PCIe Link Ctrl	RO - 0x0	
	B2	PCIe Link Status	RO - 0x0	
	C4	PCIe Dev Cap 2	RO - as PF	
	C8	PCIe Dev Ctrl 2	RO - 0x0	The Timeout value and Timeout disable of the PF are used for all VFs.
	D0	PCIe Link Ctrl 2	RO - 0x0	
	D2	PCIe Link Status 2	RO - 0x0	

Table 9-14. VF PCIe Configuration Space (Continued)

Section	Offset	Name	VF Behavior	Notes
AER Capability	100	AER - Header	RO - 0x15010001	Points to ARI structure
	104	AER - Uncorr Status	per VF	See Section 9.7.2.3.1
	108	AER - Uncorr Mask	RO - 0x0	
	10C	AER - Uncorr Severity	RO - 0x0	
	110	AER - Corr Status	per VF	See Section 9.7.2.3.2
	114	AER - Corr Mask	RO - 0x0	
	118	AER - Cap/Ctrl	per VF	Same structure as in PF
	11C:128	AER - Error Log	one log per VF	Same structure as in PF.
ARI Capability	150	ARI - Header	0x0001000E	Last
	154	ARI - Cap/Ctrl	RO - 0	

9.7.1 Legacy Header Details

9.7.1.1 VF Command Register (0x4; RW)

Bit(s)	R/W	Initial Value	Description
0	RO	0b	IOAE - I/O Access Enable RO as a zero field.
1	RO	0b	MAE - Memory Access Enable RO as a zero field.
2	RW	0b	BME - Bus Master Enable Disabling this bit prevents the associated VF from issuing any memory or I/O requests. Note that as MSI/MSI-X interrupt messages are in-band memory writes, disabling the bus master enable bit disables MSI/MSI-X interrupt messages as well. Requests other than memory or I/O requests are not controlled by this bit. **Note:** The state of active transactions is not specified when this bit is disabled after being enabled. The 82576 can choose how it behaves when this condition occurs. Software cannot count on the 82576 retaining state and resuming without loss of data when the bit is re-enabled. Transactions for a VF that has its *Bus Master Enable* bit set must not be blocked by transactions for VFs that have their *Bus Master Enable* bit cleared.
3	RO	0b	SCM - Special Cycle Enable Hardwired to 0b.
4	RO	0b	MWIE - MWI Enable Hardwired to 0b.
5	RO	0b	PSE - Palette Snoop Enable Hardwired to 0b.
6	RO	0b	PER - Parity Error Response Zero for VFs.
7	RO	0b	WCE - Wait Cycle Enable Hardwired to 0b.
8	RO	0b	SERRE - SERR# Enable Zero for VFs.
9	RO	0b	FB2BE - Fast Back-to-Back Enable Hardwired to 0b.
10	RO	0b	INTD - Interrupt Disable Hardwired to 0b.
15:11	RO	0x0	Reserved

9.7.1.2 VF Status Register (0x6; RW)

Bits	R/W	Initial Value	Description
2:0	RO	000b	Reserved
3	RO	0b	Interrupt Status Hardwired to 0b.
4	RO	1b	New Capabilities Indicates that the 82576 VFs implement extended capabilities. The 82576 VFs implement a capabilities list to indicate that it supports enhanced message signaled interrupts and PCIe extensions.
5	RO	0b	66MHz Capable Hardwired to 0b.
6	RO	0b	Reserved

Bits	R/W	Initial Value	Description
7	RO	0b	Fast Back-to-Back Capable Hardwired to 0b.
8	R/W1C	0b	MPERR - Data Parity Reported
10:9	RO	00b	DEVSEL Timing Hardwired to 0b.
11	R/W1C	0b	STA - Signaled Target Abort
12	R/W1C	0b	RTA - Received Target Abort
13	R/W1C	0b	RMA - Received Master Abort
14	R/W1C	0b	SSERR - Signaled System Error
15	R/W1C	0b	DSERR - Detected Parity Error

9.7.2 VF Legacy Capabilities

9.7.2.1 VF MSI-X Capability

The only register with a different layout than the PF for MSI-X, is the control register.

9.7.2.1.1 VF MSI-X Control Register (0x72; RW)

Bits	R/W	Initial Value	Description
10:0	RO	0x002[1]	TS - Table Size
13:11	RO	000b	Reserved
14	RW	0b	Mask - Function Mask
15	RW	0b	En - MSI-X Enable

[1] Default value is read from EEPROM

9.7.2.2 VF PCIe Capability Registers

The device control and device status registers have some fields that are specific per VF.

9.7.2.2.1 VF Device Control Register (0xA8; RW)

Bits	R/W	Default	Description
0	RO	0b	Correctable Error Reporting Enable Zero for VFs.
1	RO	0b	Non-Fatal Error Reporting Enable Zero for VFs.
2	RO	0b	Fatal Error Reporting Enable Zero for VFs.
3	RO	0b	Unsupported Request Reporting Enable Zero for VFs.
4	RO	0b	Enable Relaxed Ordering Zero for VFs.

Bits	R/W	Default	Description
7:5	RO	0b	Max Payload Size Zero for VFs.
8	RO	0b	Extended Tag field Enable Not implemented in the 82576.
9	RO	0b	Phantom Functions Enable Not implemented in the 82576.
10	RO	0b	Auxiliary Power PM Enable Zero for VFs.
11	RO	0b	Enable No Snoop Zero for VFs.
14:12	RO	000b	Max Read Request Size Zero for VFs.
15	RW	0b	Initiate Function Level Reset Specific to each VF.

9.7.2.2.2 VF Device Status Register (0xAA; RW1C)

Bits	R/W	Default	Description
0	RW1C	0b	Correctable Detected Indicates status of correctable error detection.
1	RW1C	0b	Non-Fatal Error Detected Indicates status of non-fatal error detection.
2	RW1C	0b	Fatal Error Detected Indicates status of fatal error detection.
3	RW1C	0b	Unsupported Request Detected Indicates that the 82576 received an unsupported request. This field is identical in all functions. The 82576 cannot distinguish which function caused an error.
4	RO	0b	Aux Power Detected Zero for VFs.
5	RO	0b	Transaction Pending Specific per VF. When set, indicates that a particular function (PF or VF) has issued non-posted requests that have not been completed. A function reports this bit cleared only when all completions for any outstanding non-posted requests have been received.
15:6	RO	0x00	Reserved

9.7.2.3 VF Advanced Error Reporting Registers

The following registers in the AER capability have a different behavior in a VF function.

9.7.2.3.1 VF Uncorrectable Error Status Register (0x104; R/W1CS)

Bit Location	R/W	Default Value	Description
3:0	RO	0000b	Reserved
4	RO	0b	Data Link Protocol Error Status
5	RO	0b	Surprise Down Error Status (Optional)
11:6	RO	0x0	Reserved
12	R/W1CS	0b	Poisoned TLP Status

13	RO	0b	Flow Control Protocol Error Status
14	R/W1CS	0b	Completion Timeout Status
15	R/W1CS	0b	Completer Abort Status
16	R/W1CS	0b	Unexpected Completion Status
17	RO	0b	Receiver Overflow Status
18	RO	0b	Malformed TLP Status
19	RO	0b	ECRC Error Status
20	R/W1CS	0b	Unsupported Request Error Status When caused by a function that claims a TLP.
21	RO	0b	ACS Violation Status
31:21	RO	0x0	Reserved

The Correctable Error Status register reports error status of individual correctable error sources on a PCIe device. When an individual error status bit is set to 1b, it indicates that a particular error occurred; software can clear an error status by writing a 1b to the respective bit. See the table below.

9.7.2.3.2 VF Correctable Error Status Register (0x110; R/W1CS)

Bit Location	R/W	Default Value	Description
0	RO	0b	Receiver Error Status
5:1	RO	0x0	Reserved
6	RO	0b	Bad TLP Status
7	RO	0b	Bad DLLP Status
8	RO	0b	REPLAY_NUM Rollover Status
11:9	RO	000b	Reserved
12	RO	0b	Replay Timer Timeout Status
13	R/W1CS	0b	Advisory Non-Fatal Error Status
31:14	RO	0x0	Reserved

§ §

10. Manageability

10.1 Platform Configurations

This section describes the hardware configurations for platform management. It describes the partitioning of platform manageability among system components and the functionality provided by the 82576 in each of the platform configurations.

The 82576 interfaces with an on-board BMC component for basic manageability functionality. The link between the 82576 and the BMC is either NC-SI, or SMB.

Note: Link configuration is loaded from the “Redirection LAN Interface” EEPROM bits in the Manageability Capability/Manageability Enable word (see Section 6.5.8).

10.1.1 On-Board BMC Configurations

Table 10-1Table 10-2 depicts an SMBus-only connection between the 82576 and the BMC. All communication between the 82576 and the BMC (pass-through traffic, configuration, and status) pass through a single interface. The protocol details for this configuration are according to the SMBus commands described in Section 10.2.2.

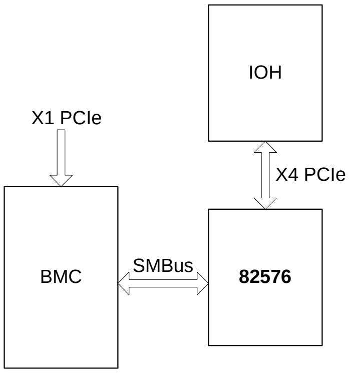


Figure 10-1. Intel® 82576 GbE Controller to BMC Connectivity Through SMBus Link

Figure 10-2 depicts an NC-SI-only connection between the 82576 and the BMC. All communication between the 82576 and the BMC (pass-through traffic, configuration, and status) pass through a single interface. The protocol details for this configuration are according to the DMTF NC-SI protocol.

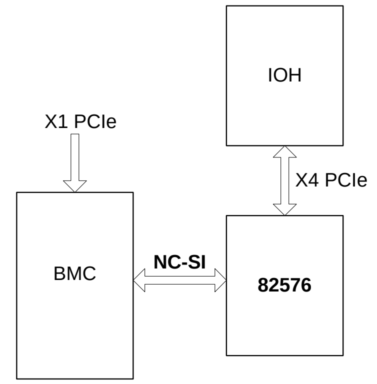


Figure 10-2. 82576 to BMC Connectivity Through a Single NC-SI Link

See next sections for a description of the traffic types that use the NC-SI and/or SMBus interfaces.

10.1.2 The 82576 NIC

The 82576 NIC do not support pass through traffic to the BMC.

10.2 Pass Through Functionality

The 82576 supports traffic pass through to an external BMC. The usable bandwidth for either direction is up to100 Mb/s in NC-SI mode & up to 400Kbps in SMBus mode. The following list describes the different flows that can make up pass through traffic:

- Intel® Active Management Technology (Intel® AMT) management traffic (low data rate)
- Keyboard or mouse traffic for KVM (low data rate)
- Video traffic for KVM (low average rate of 150 Kbytes/s to 200 Kbytes/s) - transmit only
- USB 2.0 redirect (up to 50 Mb/s)
- IDE redirect for remote CD/floppy (rate - priority 1 - CDx7 = 1.05 Mb/s. Priority 2 - CDx24 = 64 Mb/s
- Serial Over LAN (SoL) - 300 Kbytes/s

Pass through traffic is carried through an NC_SI interface or SMBus (legacy devices) based on the configuration loaded from the EEPROM. The management interface does not change dynamically and is loaded only during an the 82576 power up.

10.2.1 DMTF NC-SI Mode

The 82576 supports all the mandatory features of the DMTF NC-SI spec rev 1.0a.

10.2.1.1 Supported Feature

Table 10-1 describes the level of support of Kawela for NC-SI standard commands.

Table 10-1. Standard NC-SI Commands Support

Command	Supported
Clear initial state	Yes
Get Version ID	Yes
Get Parameters	Yes
Get Controller Packet Statistics	Yes, partially
Get Link Status	Yes[1]
Enable Channel	Yes
Disable Channel	Yes
Reset Channel	Yes
Enable VLAN	Yes[2]
Disable VLAN	Yes
Enable BCast	Yes
Disable BCast	Yes
Set MAC Address	Yes
Clear MAC Address	Yes
Get NC-SI Statistics	Yes, partially
Enable NC-SI Flow-Control	No
Disable NC-SI Flow-Control	No
Set Link Command	Yes[1,3]
Enable Global MCast Filter	Yes
Disable Global MCast Filter	Yes
Get Capabilities	Yes
Set VLAN Filters	Yes
AEN Enable	Yes
Get Pass-Through Statistics	Yes, partially
Select Package	Yes
Deselect Package	Yes
Enable Channel Network TX	Yes
Disable Channel Network TX	Yes
OEM Command	Yes

[1] When working with SGMII interface, this command is not supported.

[2] The 82576 does not support filtering of User priority/CFI Bits of VLAN

[3] In cases that on of the LAN devices is assigned for the sole use of the manageability and its LAN PCI-E function is disabled, using the NC-SI Set Link command while advertising multiple speeds and enabling Auto-Negotiation, will result in the lowest possible speed chosen. To enable link of higher a speed, the BMC should not advertise speeds that are below the desired link speed. When doing it, changing the power state of the LAN device will have not effect and the link speed will not be re-negotiated.

Table 10-2 describes the optional NC-SI features supported by the 82576.

Table 10-2. Optional NC-SI Features Support

Feature	Supported	Details
AENs	Yes.	Note: The Driver state AEN may be emitted up to 15 sec. after actual driver change.
Get Controller Packet Statistics command	No	Supports the following counters: 1-10,12-21,22-35,37-38
Get NC-SI statistics command	Yes, partially	Supports the following counters: 1-4, 7
Get NC-SI Pass-Through statistics command	Yes, partially	Supports the following counters: 2 Supports the following counters only when the OS is down: 1, 6, 7
VLAN modes	Yes, partially	Supports only modes 1,3
Buffering capabilities	Yes	8K
MAC Address filters	Yes	Supports 2 MAC addresses as mixed per port
Channel count	Yes	Supports 2 channels
VLAN filters	Yes	Supports 8 VLAN filters per port
Broadcast filters	Yes	Supports the following filters: • ARP • DHCP • NetBIOS
Multicast filters	Yes	Supports the following filters[1]: • IPv6 Neighbor Advertisement • IPv6 Router Advertisement • DHCPv6 relay and server multicast
NC-SI Flow Control command	No	
HW arbitration	Yes	Supports NC-SI HW arbitration

[1] Supports only when all three filters are enabled.

10.2.2 SMBus Pass Through (PT) Functionality

In addition to carrying pass through traffic, in SMBus mode, the 82576 provides the following manageability services to the BMC:

- ARP handling – The 82576 can be programmed to auto-ARP replying for ARP request packets and sending gratuitous ARP to reduce the traffic over the BMC interconnect.
- Teaming and fail-over – The 82576 can be configured to one of several teaming and fail-over configurations:
  - No-teaming – The 82576 dual LAN ports act independently of each other. As a result, no fail-over is supported. The BMC is responsible for teaming and fail-over.
  - Teaming – The 82576 can be configured to provide fail-over capabilities, such that manageability traffic is routed to an active port if any of the ports fail. Several modes of operation are supported.
- Default configuration of filters by EEPROM - When working in SMBus mode, the default values of the manageability receive filters can be set according to the PT LAN and flex TCO EEPROM structures defined in Section 6.7 and Section 6.9 respectively.

10.2.2.1 Pass Through (PT) Modes

PT configuration depends on how the LAN ports are configured. If the LAN ports are configured as two different channels (non-teaming mode) then the 82576 is presented on the manageability link as two different devices (via different SMBus addresses) on which each device is connected to a different LAN port. In this mode (the same as in the LAN channels), there is no logical connection between the devices. In this mode, the fail-over between the LAN ports are done by the external BMC (by sending/ receiving packets through different devices). The status reports to the BMC, ARP handling, DHCP and other pass through functionality are unique for each port.

When the 82576 operates in teaming mode, it presents itself on the SMBus a single device (one SMBus address). In this mode, the external BMC is not aware that there are two LAN ports. The 82576 determines how to route the packets it receives on the manageability channel according to the fail-over algorithm. The status reports to the BMC and other pass through configurations are common to both ports.

In pass through mode most of the manageability traffic is handled by the BMC. However, portion of the network traffic can be offloaded and by the 82576 as described in the following subsections. This configuration can be done by issuing configuration commands over the SMBus channel or the 82576 can load it from its EEPROM at power up (or both).

10.2.2.2 LAN Fail-Over in LAN Teaming Mode

Manageability fail-over is the ability in a dual-port network device to detect that the LAN connection on the manageability enabled port is lost, and to enable the other port in the device to receive/transmit manageability packets. When the 82576 operates in teaming mode, the O/S and the external BMC consider Intel® 82576 GbE Controller as one logical network device. The decision which of the the 82576 ports to use is done internally in the 82576 (or in the ANS driver in case of the regular receive/ transmit traffic). This section deals with fail-over in teaming mode only. In non-teaming mode, the external BMC should consider the 82576's network ports as different network devices, and is solely responsible for the fail-over mechanism.

10.2.2.2.1 Fail-Over Functionality

In teaming mode, the 82576 mirrors both the network ports into a single SMBus slave device. The 82576 will automatically handle the configurations of both network ports. Thus, for configurations, receiving & transmitting the BMC should consider both ports as a single entity.

When the currently active port for transmission becomes unavailable (i.e. the link is down), the 82576 will automatically try to switch the packet transmission to another port. Thus, as long as one of the ports is valid, the BMC will have a valid link indication for the SMBus slave.

10.2.2.2.1.1 Transmit functionality

In order to transmit a packet the BMC should issue the appropriate SMBus packet transmission commands to the 82576. The 82576 will then automatically choose the transmission port.

10.2.2.2.1.2 Receive Functionality

When the 82576 receives a packet from any of the teamed ports it will notify and forward the packet to the BMC.

Note: As both ports might be active (i.e. with a valid link) packets may be received on the currently non-active port. To avoid this, packet duplication, failover should not be enabled when connected to a Hub.

Note: Fail over and teaming are not supported in NC-SI mode.

10.2.2.2.1.3 Port Switching (Fail-Over)

While in teaming mode, transmit traffic is always transmitted by the 82576 through only one of the ports at any given time. The 82576 may switch the traffic transmission between ports under any of the following conditions:

1. The current transmitting port link is not available
2. The preferred primary port is enabled and becomes available for transmission.

10.2.2.2.1.4 Driver Interactions

When the LAN driver is present, the decision to switch between the ports is done by the driver. When the driver is absent, this decision is done internally by the 82576.

Note: When the driver releases teaming mode (e.g. when system state changes), the 82576 reconfigure the LAN ports to teaming mode. The 82576 accomplishes this by re-setting the MAC address of the ports to be the teaming address in order to re-start teaming. This is followed by transmission of gratuitous ARP packets to notify the network of teaming mode re-setting.

10.2.2.2.2 Fail-Over Configuration

Fail-over operation is configured through the “fail-over configuration register”, as described in Table 10-3

The BMC should configure this register after every “initialization indication” from the 82576 (after every the 82576 firmware reset).

The different configurations available to the BMC are detailed in this section.

Note: In teaming mode both ports should be configured with the same receive manageability filters parameters (EEPROM sections for both ports should be identical).

10.2.2.2.2.1 Preferred Primary Port

The BMC may choose one of the network ports as a preferred primary port for packet transmission. Intel® 82576 GbE Controller will use the preferred primary port as the transmission port whenever the link for that port is valid. I.e. Intel® 82576 GbE Controller will always switch back to the preferred primary port when available.

10.2.2.2.2.2 Gratuitous ARPs

In order to notify the link partner that a port switching has occurred, Intel® 82576 GbE Controller may be configured to automatically send gratuitous ARPs. These gratuitous ARPs will cause the link partner to update its ARP tables to reflect the change.

The BMC may enable/disable gratuitous ARPs, configure the number of gratuitous ARPs or the interval between them by modifying the Fail Over configuration register.

10.2.2.2.2.3 Link Down Timeout

The BMC can control the timeout for a link to be considered invalid. Intel® 82576 GbE Controller will wait this timeout before attempting to switch from an inactive port.

10.2.2.2.3 Fail-Over Register

This register is loaded on power up from the EEPROM (see Section 6.8.5 and Section 6.8.6), or through the “set fail-over configuration” host command by the LAN driver (see Section 10.3.3.9). The bits in this register can also be changed by Intel® 82576 GbE Controller HW to present its current state.

Table 10-3 describes the register bits.

Table 10-3. Failover Configuration Register

Bit(s)	Field	Initial Value	Read / Write	Description
0	RMP0EN	0x1	RO	RCV MNG port 0 Enable. When this bit is set, it reports that MNG traffic will be received from port 0.
1	RMP1EN	0x1	RO	RCV MNG port 1 Enable. When this bit is set, it reports that MNG traffic will be received from port 1.
2	MXP	0x0	RO	MNG XMT Port. 0b = reports that MNG traffic should be transmitted through port 0. 1b = reports that MNG traffic should be transmitted through port 1.
3	PRPP	0x0	RW	Preferred Primary Port. 0b = Port 0 is the preferred primary port. 1b = Port 1 is the preferred primary port.
4	PRPPE	0x0	RW	Preferred Primary Port enables.
5	Reserved	0x0	RO	Reserved
6	RGAEN	0x0	RW	Repeated Gratuitous ARP Enable. If this bit is set, Intel® 82576 GbE Controller sends a configurable number of Gratuitous ARP packets (GAC bits of this register) using configurable interval (GATI bits of this register) after the following events: System move to Dx, Fail-over event initiated Intel® 82576 GbE Controller
8:7	Reserved	0x0	0	Reserved
9	TFOENODX	0x0	RW	Teaming Fail Over Enable on Dx. Enable Fail Over mechanism. Bits 3-8 are valid only if this bit is set.
11:10	Reserved	0x0	RO	Reserved
12-15	GAC	0x0	RW	Gratuitous ARP counter. Indicates the number of Gratuitous ARP that should be sent after a failover event and after move to Dx. The value 0b means that there is no limit on the Gratuitous ARP packets.
16-23	LDFOT	0x0	RW	Link down Fail Over time. Defines the time in seconds the link should be down before doing a fail over to the second port. This is also the time that the primary link should be up (after it was down) before Intel® 82576 GbE Controller will fail over back to the primary port.
24-31	GATI	0x0	RW	Gratuitous ARP Transmission Interval. Defines the interval in seconds before retransmission of Gratuitous ARP packets.

10.2.2.3 ARP Handling

Independent of the management interface, the 82576 might be programmed by the BMC to provide ARP services. The 82576 supports auto-ARP replying for ARP request packets and sending Gratuitous ARP. Auto-ARP is done in both ports in both modes (dual- and one-channel mode). In dual-channel mode, each channel uses its own IP and MAC address (either the operating system MAC address or independent addresses). In one-channel mode, both ports use the same IP and MAC address and the ARP is responded to through the port it was received.

Note: When configured in NC-SI mode, the 82576 does not provide ARP services. All ARP handling is done by the BMC.

10.2.2.3.1 ARP Operation

ARP parameters are loaded from the EEPROM when at the 82576 power up or configured through the management interface. These parameters should be configured to enable ARP operation:

- ARP auto-reply enabled
- ARP IP address (to filter ARP packets)
- ARP MAC addresses (for ARP response)

When an ARP request packet is received on the wire and ARP auto-reply is enabled, the 82576 checks the targeted IP address (after the packet has passed L2 checks and ARP checks). If the targeted IP matches the 82576 IP configuration, then it replies with an ARP response. The 82576 responds to the ARP request targeted to the ARP IP address with the ARP MAC address configured to it. In a case where there is no match, the 82576 silently discards the packets. If the 82576 is not configured to do auto-ARP response, it forwards the ARP packets to the BMC.

When the external BMC uses the same IP and MAC of the operating system, the ARP operation should be coordinated with the operating system operation. In this mode, the external BMC has the responsibility of ARP generation and “ARP auto-reply” should be disabled.

10.2.2.3.2 ARP Packet Formats

Table 10-4. ARP Request Packet

Offset	# Of bytes	Field	Value (In Hex)	Action
0	6	Destination Address		Compare
6	6	Source Address		Stored
12	8	Possible LLC/SNAP Header		Stored
12	4	Possible VLAN Tag		Stored
12	2	Type	0806	Compare
14	2	HW Type	0001	Compare
16	2	Protocol Type	0800	Compare
18	1	Hardware Size	06	Compare
19	1	Protocol Address Length	04	Compare
20	2	Operation	0001	Compare
22	6	Sender HW Address	-	Stored
28	4	Sender IP Address	-	Stored
32	6	Target HW Address	-	Ignore
38	4	Target IP Address	ARP IP address	Compare

Table 10-5. ARP Response Packet

Offset	# of bytes	Field	Value
0	6	Destination Address	ARP Request Source Address
6	6	Source Address	Programmed from EEPROM or BMC
12	8	Possible LLC/SNAP Header	From ARP Request
12	4	Possible VLAN Tag	From ARP Request
12	2	Type	0x0806
14	2	HW Type	0x0001
16	2	Protocol Type	0x0800
18	1	Hardware Size	0x06
19	1	Protocol Address Length	0x04
20	2	Operation	0x0002
22	6	Sender HW Address	Programmed from EEPROM or BMC
28	4	Sender IP Address	Programmed from EEPROM or BMC
32	6	Target HW Address	ARP Request Sender HW Address
38	4	Target IP Address	ARP Request Sender IP Address

Table 10-6. Gratuitous ARP Packet

Offset	# of bytes	Field	Value
0	6	Destination Address	Broadcast Address
6	6	Source Address	
12	2	Type	0x0806
14	2	HW Type	0x0001
16	2	Protocol Type	0x0800
18	1	Hardware Size	0x06
19	1	Protocol Address Length	0x04
20	2	Operation	0x0001
22	6	Sender HW Address	
28	4	Sender IP Address	
32	6	Target HW Address	
38	4	Target IP Address	

10.3 Programming Interfaces

10.3.1 NC-SI Programming

In addition to the regular NC-SI commands described in Table 10-1 the following Intel vendor specific commands are supported. The purpose of these commands is to provide a means for the Management Controller (BMC) to access some of the Intel specific features present in the 82576.

10.3.1.1 Method

The following features are abstracted via NC-SI using the OEM specific command:

- RX filters:
  - Packet Addition Decision Filters 0x0...0x4
  - Packet Reduction Decision Filters 0x5...0x7
  - MNG2HOST register (Controls the forwarding of manageability packets to the host)
  - Flex 128 filters 0x0...0x3
  - Flex TCP/UDP port filters 0..0xA
  - IPv4/IPv6 filters
  - Ether type filters
- Get System MAC Address – This command allows the BMC to retrieve the System MAC address used by the NC. This MAC address may be used for a “Shared MAC Address” mode.
- “Keep PHY Link Up” (Veto bit) Enable/Disable – This feature allows the BMC to block PHY reset, which might cause session loss.
- TCO Reset – Allows the Management Controller to reset the Network Adapter.
- Checksum offloading – offloads IP/UDP/TCP checksum checking from the Management Controller.

These commands are designed to be compliant with their corresponding SMBus commands (if existing).

All of the commands are based on a single DMTF defined NC-SI command, known as “OEM Command”. This command is detailed in Section 10.3.1.2.

10.3.1.2 OEM Command (0x50)

The OEM commands is used to expose additional capabilities of the 82576 to the BMC.

The Vendor Enterprise Number is the unique MIB/SNMP Private Enterprise number assigned by IANA per organization. Intel devices use a Manufacturer ID of 0x157.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..	Intel Command Number	Optional Data		

10.3.1.3 OEM Response (0xD0)

The sideband interface shall return Unknown Command Type reason code for any un-recognized enterprise number using the following frame format. If the command is valid, the response, if any, is allowed to be vendor-specific. It is recommended to use the 0x8000 range for vendor-specific code.

Note: Responses have no command-specific reason code, unless otherwise specified within the command.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			

16..19	Response Code	Reason Code
20..23	Manufacturer ID (Intel 0x157)	
24..27	Intel Command Number	Optional Return Data

Table 10-7. OEM Specific Command Response and Reason Codes

Response Code		Reason Code	
Value	**Description**	**Value**	**Description**
0x1	Command Failed	0x5081	Invalid Intel Command Number
		0x5082	Invalid Intel Command Parameter Number

10.3.1.4 Command Summary

Table 10-8. Command Summary

Intel Command	Parameter	Command name
0x00	0x00	Set IP Filters control
0x01	0x00	Get IP Filters control
0x02	0x0A	Set Manageability to Host
	0x10	Set Flexible 128 Filter 0 Mask & length
	0x11	Set Flexible 128 Filter 0 Data
	0x20	Set Flexible 128 Filter 1 Mask & length
	0x21	Set Flexible 128 Filter 1 Data
	0x30	Set Flexible 128 Filter 2 Mask & length
	0x31	Set Flexible 128 Filter 2 Data
	0x40	Set Flexible 128 Filter 3 Mask & length
	0x41	Set Flexible 128 Filter 3 Data
	0x61	Set Packet Addition Filters
	0x63	Set Flex TCP/UDP Port filters
	0x64	Set Flex IPv4 Address filters
	0x65	Set Flex IPv6 Address filters

Table 10-8. Command Summary (Continued)

0x3	0x0A	Get Manageability to Host
	0x10	Get Flexible 128 Filter 0 Mask & length
	0x11	Get Flexible 128 Filter 0 Data
	0x20	Get Flexible 128 Filter 1 Mask & length
	0x21	Get Flexible 128 Filter 1 Data
	0x30	Get Flexible 128 Filter 2 Mask & length
	0x31	Get Flexible 128 Filter 2 Data
	0x40	Get Flexible 128 Filter 3 Mask & length
	0x41	Get Flexible 128 Filter 3 Data
	0x61	Get Packet Addition Filters
	0x63	Get Flex TCP/UDP Port filters
	0x64	Get Flex IPv4 Address filters
	0x65	Get Flex IPv6 Address filters
0x04	0x00	Set Unicast Packet Reduction
	0x01	Set Multicast Packet Reduction
	0x02	Set Broadcast Packet Reduction
0x05	0x00	Get Unicast Packet Reduction
	0x01	Get Multicast Packet Reduction
	0x02	Get Broadcast Packet Reduction
0x06	N/A	Get System MAC Address
0x20	N/A	Set Intel Management Control
0x21	N/A	Get Intel Management Control
0x22	N/A	Perform TCO Reset
0x23	N/A	Enable IP/UDP/TCP Checksum offloading
0x24	N/A	Disable IP/UDP/TCP Checksum offloading

10.3.1.5 Set Intel Filters Control Command (Intel Command 0x00)

10.3.1.5.1 Set Intel Filters Control – IP Filters Control Command (Intel Command 0x00, Filter Control Index 0x00)

This command controls different aspects of the Intel Filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x00	0x00	IP Filters control (3-2)	
24..27	IP Filters Control (1-0)			

Where “IP Filters Control” has the following format:

Table 10-9. IP Filter Formats

Bit #	Name	Description	Default value
0	IPv4/IPv6 Mode	IPv6 (0b): There are 0 IPv4 filters and 4 IPv6 filters IPv4 (1b): There are 4 IPv4 filters and 3 IPv6 filters See Section 8.21.9 or Section 10.4.3.6 for details.	1b
1..15	Reserved		
16	IPv4 Filter 0 Valid	Indicates if the IPv4 address configured in IPv4 address 0 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv4 Filter Command” is used for filter 0.
17	IPv4 Filter 1 Valid	Indicates if the IPv4 address configured in IPv4 address 1 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv4 Filter Command” is used for filter 1.
18	IPv4 Filter 2 Valid	Indicates if the IPv4 address configured in IPv4 address 2 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv4 Filter Command” is used for filter 2.
19	IPv4 Filter 3 Valid	Indicates if the IPv4 address configured in IPv4 address 3 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv4 Filter Command” is used for filter 3.
20..23	Reserved		
24	IPv6 Filter 0 Valid	Indicates if the IPv6 address configured in IPv6 address 0 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv6 Filter Command” is used for filter 0.
25	IPv6 Filter 1 Valid	Indicates if the IPv6 address configured in IPv6 address 1 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv6 Filter Command” is used for filter 1.
26	IPv6 Filter 2 Valid	Indicates if the IPv6 address configured in IPv6 address 2 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv6 Filter Command” is used for filter 2.
27	IPv6 Filter 3 Valid	Indicates if the IPv6 address configured in IPv6 address 3 is valid.	0b Note: The Network Controller shall automatically set this bit to 1b if the “Set Intel Filter – IPv6 Filter Command” is used for filter 3.
28..31	Reserved		

10.3.1.5.2 Set Intel Filters Control – IP Filters Control Response (Intel Command 0x00, Filter Control Index 0x00)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x00	0x00		

10.3.1.6 Get Intel Filters Control Command (Intel Command 0x01)

10.3.1.6.1 Get Intel Filters Control – IP Filters Control Command (Intel Command 0x01, Filter Control Index 0x00)

This command controls different aspects of the Intel Filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x01	0x00		

10.3.1.6.2 Get Intel Filters Control – IP Filters Control Response (Intel Command 0x01, Filter Control Index 0x00)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x01	0x00	IP Filters Control (3-2)	
28..29	IP Filters Control (1-0)			

IP Filter Control: See Table 10-9.

10.3.1.7 Set Intel Filters Formats

10.3.1.7.1 Set Intel Filters Command (Intel Command 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**

00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x02	Filter Parameter	Filters Data (optional)	

10.3.1.7.2 Set Intel Filters Response (Intel Command 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..	0x02	Filter Parameter	Return Data (Optional)	

10.3.1.7.3 Set Intel Filters – Manageability to Host Command (Intel Command 0x02, Filter parameter 0x0A)

This command sets the Mng2Host register. The Mng2Host register controls whether Pass-Through packets destined to the BMC will also be forwarded to the host OS.

The Mng2Host register has the following structure:

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x0A	Manageability to Host (3-2)	
24..25	Manageability to Host (1-0)			

Table 10-10. Manageability to Host field

Bits	Name	Description	Default
0	Decision Filter 0	Determines if packets that have passed Decision Filter 0 will be also forwarded to the host OS.	0b
1	Decision Filter 1	Determines if packets that have passed Decision Filter 1 will be also forwarded to the host OS.	0b
2	Decision Filter 2	Determines if packets that have passed Decision Filter 2 will be also forwarded to the host OS.	0b
3	Decision Filter 3	Determines if packets that have passed Decision Filter 3 will be also forwarded to the host OS.	0b
4	Decision Filter 4	Determines if packets that have passed Decision Filter 4 will be also forwarded to the host OS.	0b
5	Unicast & Mixed	Determines if Unicast & Mixed packets will be also forwarded to the host OS.	0b

Table 10-10. Manageability to Host field

6	Global Multicast	Determines if Global Multicast packets will be also forwarded to the host OS.	1b
7	Broadcast	Determines if Broadcast packets will be also forwarded to the host OS.	1b
31:8	Reserved	Reserved	N/A

10.3.1.7.4 Set Intel Filters – Manageability to Host Response (Intel Command 0x02, Filter parameter 0x0A)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..	0x02	0x0A		

10.3.1.7.5 Set Intel Filters – Flex Filter 0/1/2/3 Enable Mask and Length Command (Intel Command 0x02, Filter parameter 0x10/0x20/0x30/0x40)

The following command sets the Intel Flex filters Mask and length. Use Filter parameters 0x10/0x20/0x30/0x40 for Flexible filters 0/1/2/3 accordingly. See Section 10.4.3.5 for details of the programming.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x10/ 0x20/ 0x30/ 0x40	Mask Byte 1	Mask Byte 2
24..27	..	..	..	..
28...31	..	..	..	..
32...35	..	..	..	..
35...37	..	Mask Byte 16	Reserved	Reserved
38	Flexible Filter Length (8-128 bytes)			

10.3.1.7.6 Set Intel Filters – Flex Filter 0/1/2/3 Enable Mask and Length Response (Intel Command 0x02, Filter parameter 0x10/0x20/0x30/0x40)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			

16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x10/ 0x20/ 0x30/ 0x40		

10.3.1.7.7 Set Intel Filters – Flex Filter 0/1/2/3 Data Command (Intel Command 0x02, Filter parameter 0x11/0x21/0x31/0x41)

The following command sets the Intel Flex filters Data. Use Filter parameters 0x11/0x21/0x31/0x41 for Flexible filters 0/1/2/3 accordingly.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..	0x02	0x11/ 0x21/ 0x31/ 0x41	Filter Data Group	Filter Data 1
	..	Filter Data N		

The Filter Data Group parameter defines which bytes of the Flex filter are set by this command:

Table 10-11. Filter Data Group

Code	Bytes programmed	Filter Data Length
0x0	bytes 0-29	1 - 30
0x1	bytes 30-59	1 - 30
0x2	bytes 60-89	1 - 30
0x3	bytes 90-119	1 - 30
0x4	bytes 120-127	1 - 8

10.3.1.7.8 Set Intel Filters – Flex Filter 0/1/2/3 Data Response (Intel Command 0x02, Filter parameter 0x11/0x21/0x31/0x41)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x11/ 0x21/ 0x31/ 0x41		

Note: If Filter Data Length is larger than specified in Table 10-11 an Out of Range Reason code is returned.

10.3.1.7.9 Set Intel Filters – Packet Addition Decision Filter Command (Intel Command 0x02, Filter parameter 0x61)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x61	Filter index	Decision Filter (MSB)
24..26	.....		Decision Filter (LSB)	

Filter index range: 0x0..0x4

If the filter index is bigger than 4, a command failed Response Code is returned with no reason.

Table 10-12. Filter Values

Bit #	Name	Description
0	Unicast (AND)	If set, packets must match a Unicast filter
1	Broadcast (AND)	If set, packets must match the Broadcast filter
2	VLAN (AND)	If set, packets must match a VLAN filter
3	IP Address (AND)	If set, packets must match an IP filter
4	Unicast (OR)	If set, packets must match a Unicast filter or a different “OR” filter
5	Broadcast	If set, packets must match the Broadcast filter or a different “OR” filter
6	Multicast (AND)	If set, packets must match the Multicast filter
7	ARP Request (OR)	If set, packets must match the ARP Request filter or a different OR filter
8	ARP Response (OR)	If set, packets can pass if match the ARP Response filter
9	Neighbor Discovery (OR)	If set, packets can pass if match the Neighbor Discovery filter
10	Port 0x298 (OR)	If set, packets can pass if match a fixed TCP/UDP Port 0x298 filter
11	Port 0x26F (OR)	If set, packets can pass if match a fixed TCP/UDP Port 0x26F filter
12	Flex port 0 (OR)	If set, packets can pass if match the TCP/UDP Port filter 0
13	Flex port 1 (OR)	If set, packets can pass if match the TCP/UDP Port filter 1
14	Flex port 2 (OR)	If set, packets can pass if match the TCP/UDP Port filter 2
15	Flex port 3 (OR)	If set, packets can pass if match the TCP/UDP Port filter 3
16	Flex port 4 (OR)	If set, packets can pass if match the TCP/UDP Port filter 4
17	Flex port 5 (OR)	If set, packets can pass if match the TCP/UDP Port filter 5
18	Flex port 6 (OR)	If set, packets can pass if match the TCP/UDP Port filter 6
19	Flex port 7 (OR)	If set, packets can pass if match the TCP/UDP Port filter 7
20	Flex port 8 (OR)	If set, packets can pass if match the TCP/UDP Port filter 8
21	Flex port 9 (OR)	If set, packets can pass if match the TCP/UDP Port filter 9

Table 10-12. Filter Values (Continued)

22	Flex port 10 (OR)	If set, packets can pass if match the TCP/UDP Port filter 10
23	DHCPv6 (OR)	If set, packets can pass if match the DHCPv6 port (0x0223)
24	DHCP Client (OR)	If set, packets can pass if match the DHCP Server port (0x0043)
25	DHCP Server (OR)	If set, packets can pass if match the DHCP Client port (0x0044)
26	NetBIOS Name Service (OR)	If set, packets can pass if match the NetBIOS Name Service port (0x0089)
27	NetBIOS Datagram Service (OR)	If set, packets can pass if match the NetBIOS Datagram Service port (0x008A)
28	Flex TCO 0 (OR)	If set, packets can pass if match the Flex 128 TCO filter 0
29	Flex TCO 1 (OR)	If set, packets can pass if match the Flex 128 TCO filter 1
30	Flex TCO 2 (OR)	If set, packets can pass if match the Flex 128 TCO filter 2
31	Flex TCO 3 (OR)	If set, packets can pass if match the Flex 128 TCO filter 3

The filtering is divided into 2 decisions:

Bits 0,1,2,3,6 works in an “AND” manner –Thus, they all must be true in order for a packet to pass (if any were set).

Bits 5,7-31 work in an “OR” manner – Thus, at least one of them must be true for a packet to pass (if any were set).

See Section 10.4.4 for description of the decision filters.

Note: These filter settings will operate according to the VLAN mode, as configured according to the DMTF NC-SI specification. After disabling Packet Reduction filters the BMC must re-set the VLAN mode using the “Set VLAN” command.

10.3.1.7.10 Set Intel Filters – Packet Addition Decision Filter Response (Intel Command 0x02, Filter parameter 0x61)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x61		

10.3.1.7.11 Set Intel Filters – Flex TCP/UDP Port Filter Command (Intel Command 0x02, Filter parameter 0x63)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x63	Port filter index	TCP/UDP Port MSB
24	TCP/UDP Port LSB			

Filter index range: 0x0..0xA

If the filter index is bigger than 10, a command failed Response Code is returned with no reason.

10.3.1.7.12 Set Intel Filters – Flex TCP/UDP Port Filter Response (Intel Command 0x02, Filter parameter 0x63)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x63		

10.3.1.7.13 Set Intel Filters – IPv4 Filter Command (Intel Command 0x02, Filter parameter 0x64)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x64	IP filter index	IPv4 Address (MSB)
24..26	...		IPv4 Address (LSB)	

Note: The filters index range can vary according to the “IPv4/IPv6 Mode” setting in the “Filters Control” command

IPv4 Mode: Filter index range: 0x0..0x3

IPv6 Mode: This command should not be used in IPv6 mode.

10.3.1.7.14 Set Intel Filters – IPv4 Filter Response (Intel Command 0x02, Filter parameter 0x64)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x64		

If the IP filter index is bigger than 3, a command failed Response Code is returned with no reason.

10.3.1.7.15 Set Intel Filters – IPv6 Filter Command (Intel Command 0x02, Filter parameter 0x65)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x65	IP filter index	IPv6 Address (MSB, byte 15)
24..27	..	..	..	..
28..31	..	..	..	..
32..35	..	..	..	..
36..38	..	..	IPv6 Address (LSB, byte 0)	

Note: The filters index range can vary according to the “IPv4/IPv6 Mode” setting in the “Filters Control” command

IPv4 Mode: Filter index range: 0x1..0x3
IPv6 Mode: Filter index range: 0x0..0x3

10.3.1.7.16 Set Intel Filters – IPv6 Filter Response (Intel Command 0x02, Filter parameter 0x65)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x65		

If the IP filter index is bigger than 3, a command failed Response Code is returned with no reason.

10.3.1.7.17 Set Intel Filters - EtherType Filter Command (Intel Command 0x02, Filter parameter 0x67)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x67	EtherType Filter Index	EtherType Filter MSB
24..27	..	..	EtherType Filter LSB	

Where the EtherType Filter has the format as described in Section 8.21.3.

Table 10-13. Ethertype usage

Filter #	Usage	Note
0-1	Reserved	Not available for generic use
2	User defined	
3	User defined	

10.3.1.7.18 Set Intel Filters - EtherType Filter Response (Intel Command 0x02, Filter parameter 0x67)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x67		

If the Ethertype filter Index is different than 2 or 3, a command failed Response Code is returned with no reason.

10.3.1.7.19 Set Intel Filters - Packet Addition Extended Decision Filter Command (Intel Command 0x02, Filter parameter 0x68)

DecisionFilter0 Bits 5,7-31 and DecisionFilter1 bits 8..10 work in an “OR” manner – Thus, at least one of them must be true for a packet to pass (if any were set).

See Figure 10-3 for description of the decision filters structure.

The command shall overwrite any previously stored value.

Note: Previous “Set Intel Filters – Packet Addition Decision Filter” command (0x61) should be kept and supported. For legacy reasons - If previous “Decision Filter” command is called – it should set the Decision Filter 0 as provided. The extended Decision Filter remains unchanged.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x68	Extended Decision filter Index	Extended Decision filter 1 MSB
24..27	..	..	Extended Decision filter 1 LSB	Extended Decision filter 0 MSB
28..30	..	..	Extended Decision filter 0 LSB	

Extended Decision filter Index Range: 0..5

Filter 0: See Table 10-12.

Filter 1: See table below:

Table 10-14. Extended Filter 1 Values

Bit #	Name	Description
0	Ethertype 0x88F8	AND filter
1	Ethertype 0x8808	AND filter
3:2	Ethertype 2 -3	AND filters
7:4	Reserved	Reserved
8	Ethertype 0x88F8	OR filter
9	Ethertype 0x8808	OR filter
11:10	Ethertype 2 -3	OR filters
31:12	Reserved	Reserved

10.3.1.7.20 Set Intel Filters – Packet Addition Extended Decision Filter Response (Intel Command 0x02, Filter parameter 0x68)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x02	0x68		

If the Extended Decision filter Index is bigger than 5, a command failed Response Code is returned with no reason.

10.3.1.8 Get Intel Filters Formats

10.3.1.8.1 Get Intel Filters Command (Intel Command 0x03)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	Parameter Number		

10.3.1.8.2 Get Intel Filters Response (Intel Command 0x03)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x03	Parameter Number	Optional Return Data	

10.3.1.8.3 Get Intel Filters – Manageability to Host Command (Intel Command 0x03, Filter Parameter 0x0A)

This command retrieves the Mng2Host register. The Mng2Host register controls whether Pass-Through packets destined to the BMC will also be forwarded to the host OS.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	0x0A		

10.3.1.8.4 Get Intel Filters – Manageability to Host Response (Intel Command 0x03, Filter parameter 0x0A)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x0A	Manageability to Host (MSB)	
28..29	Manageability to Host (LSB)			

The Mng2Host register has the following structure.

Table 10-15. Mng2Host Structure

Bits	Description	Default
0	Decision Filter 0	Determines if packets that have passed Decision Filter 0 will be also forwarded to the host OS.
1	Decision Filter 1	Determines if packets that have passed Decision Filter 1 will be also forwarded to the host OS.
2	Decision Filter 2	Determines if packets that have passed Decision Filter 2 will be also forwarded to the host OS.
3	Decision Filter 3	Determines if packets that have passed Decision Filter 3 will be also forwarded to the host OS.
4	Decision Filter 4	Determines if packets that have passed Decision Filter 4 will be also forwarded to the host OS.

Table 10-15. Mng2Host Structure (Continued)

5	Unicast & Mixed	Determines if Unicast & Mixed packets will be also forwarded to the host OS.
6	Global Multicast	Determines if Global Multicast packets will be also forwarded to the host OS.
7	Broadcast	Determines if Broadcast packets will be also forwarded to the host OS.
31:8	Reserved	Reserved

10.3.1.8.5 Get Intel Filters – Flex Filter 0 Enable Mask and Length Command (Intel Command 0x03, Filter Parameter 0x10/0x20/0x30/0x40)

The following command retrieves the Intel Flex filters Mask and length. Use Filter parameters 0x10/0x20/0x30/0x40 for Flexible filters 0/1/2/3 accordingly. See Section 10.4.3.5 for details of the values returned by this command.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	0x10/ 0x20/ 0x30/ 0x40		

10.3.1.8.6 Get Intel Filters – Flex Filter 0 Enable Mask and Length Response (Intel Command 0x03, Filter parameter 0x10/0x20/0x30/0x40)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x10/ 0x20/ 0x30/ 0x40	Mask Byte 1	Mask Byte 2
28..31	..	..	..	..
32..35	..	..	..	..
36..39	..	..	..	..
40..43	..	Mask Byte 16	Reserved	Reserved
44	Flexible Filter Length			

10.3.1.8.7 Get Intel Filters – Flex Filter Data Command (Intel Command 0x03, Filter Parameter 0x11/0x21/0x31/0x41)

The following command retrieves the Intel Flex filters Data. Use Filter parameters 0x11/0x21/0x31/0x41 for Flexible filters 0/1/2/3 accordingly.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x11/ 0x21/ 0x31/ 0x41	Filter Data Group 0..4	

The Filter Data Group parameter defines which bytes of the Flex filter are returned by this command:

Table 10-16. Filter Data Group

Code	Bytes Returned
0x0	bytes 0-29
0x1	bytes 30-59
0x2	bytes 60-89
0x3	bytes 90-119
0x4	bytes 120-127

10.3.1.8.8 Get Intel Filters – Flex Filter Data Response (Intel Command 0x03, Filter parameter 0x11)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..	0x03	0x11/ 0x21/ 0x31/ 0x41	Filter Data Group	Filter Data 1
	..	Filter Data N		

If the filter group number is bigger than 4, a command failed Response Code is returned with no reason.

10.3.1.8.9 Get Intel Filters – Packet Addition Decision Filter Command (Intel Command 0x03, Filter Parameter 0x61)

	Bits			
Bytes	31..24	23..16	15..08	07..00

00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x61	Decision filter index	

Filter index range: 0x0..0x4

10.3.1.8.10 Get Intel Filters – Packet Addition Decision Filter Response (Intel Command 0x03, Filter parameter 0x0A)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x61	Decision Filter (MSB)	
28..29	Decision Filter (LSB)			

The Decision filter structure returned is described in Table 10-12.

If the Decision filter index is bigger than 4, a command failed Response Code is returned with no reason.

10.3.1.8.11 Get Intel Filters – Flex TCP/UDP Port Filter Command (Intel Command 0x03, Filter Parameter 0x63)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x63	TCP/UDP Filter Index	

Filter index range: 0x0..0xA

10.3.1.8.12 Get Intel Filters – Flex TCP/UDP Port Filter Response (Intel Command 0x03, Filter parameter 0x63)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	

20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x63	TCP/UDP Filter Index	TCP/UDP Port (1)
28	TCP/UDP Port (0)			

Filter index range: 0x0..0xA

If the TCP/UDP Filter Index is bigger than 10, a command failed Response Code is returned with no reason

10.3.1.8.13 Get Intel Filters – IPv4 Filter Command (Intel Command 0x03, Filter Parameter 0x64)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x64	IPv4 Filter Index	

Note: The filters index range can vary according to the “IPv4/IPv6 Mode” setting in the “Filters Control” command

IPv4 Mode: Filter index range: 0x0..0x3
IPv6 Mode: This command should not be used in IPv6 mode.

10.3.1.8.14 Get Intel Filters – IPv4 Filter Response (Intel Command 0x03, Filter parameter 0x64)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x64	IPv4 Filter Index	IPv4 Address (3)
28..29	IPv4 Address (2-0)			

If the IPv4 Filter Index is bigger than 3, a command failed Response Code is returned with no reason.

10.3.1.8.15 Get Intel Filters – IPv6 Filter Command (Intel Command 0x03, Filter Parameter 0x65)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**

00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x65	IPv6 Filter Index	

Note: The filters index range can vary according to the “IPv4/IPv6 Mode” setting in the “Filters Control” command.

IPv4 Mode: Filter index range: 0x0..0x2
IPv6 Mode: Filter index range: 0x0..0x3

10.3.1.8.16 Get Intel Filters – IPv6 Filter Response (Intel Command 0x03, Filter parameter 0x65)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x65	IPv6 Filter Index	IPv6 Address (MSB, Byte 16)
28..31	..	..	..	..
32..35	..	..	..	..
36..39	..	..	..	..
40..42	..	..	IPv6 Address (LSB, Byte 0)	

If the IPv6 Filter Index is bigger than 3, a command failed Response Code is returned with no reason.

10.3.1.8.17 Get Intel Filters - EtherType Filter Command (Intel Command 0x03, Filter parameter 0x67)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x67	EtherType Filter Index	

Valid indices: 2..3

See Table 10-13 for description of the various Ethertype filters usage.

10.3.1.8.18 Get Intel Filters - EtherType Filter Response (Intel Command 0x03, Filter parameter 0x67)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x67	EtherType Filter Index	EtherType Filter MSB
28..30	..	..	EtherType Filter LSB	

If the Ethertype filter Index is different than 2 or 3, a command failed Response Code is returned with no reason.

10.3.1.8.19 Get Intel Filters – Packet Addition Extended Decision Filter Command (Intel Command 0x03, Filter parameter 0x68)

This command allows the BMC to retrieve the Extended Decision Filter.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x68	Extended Decision Filter Index	

10.3.1.8.20 Get Intel Filters – Packet Addition Extended Decision Filter Response (Intel Command 0x03, Filter parameter 0x68)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x03	0x68	Decision Filter Index	Decision Filter 1 MSB
28..31	..	..	Decision Filter 1 LSB	Decision Filter 0 MSB
32..34	..	..	Decision Filter 0 LSB	

Where Decision Filter 0 & Decision Filter 1 have the structure as detailed in the respective “Set” commands.

If the Extended Decision Filter Index is bigger than 4, a command failed Response Code is returned with no reason.

10.3.1.9 Set Intel Packet Reduction Filters Formats

10.3.1.9.1 Set Intel Packet Reduction Filters Command (Intel Command 0x04)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20...	0x04	Packet Reduction Index	Packet Reduction Data.....	

Note: It is advised that the BMC only uses the Extended Packet Reduction commands.

The “Packet Reduction Data” field has the following structure:

Table 10-17. Packet Reduction Data

Bit #	Name	Description
2:0	Reserved	
3	IP Address	If set, all packets must also match an IP filter
9:4	Reserved	
10	Port 0x298	If set, all packets can pass if match a fixed TCP/UDP Port 0x298 filter
11	Port 0x26F	If set, all packets can pass if match a fixed TCP/UDP Port 0x26F filter
12	Flex port 0	If set, all packets can pass if match the TCP/UDP Port filter 0
13	Flex port 1	If set, all packets can pass if match the TCP/UDP Port filter 1
14	Flex port 2	If set, all packets can pass if match the TCP/UDP Port filter 2
15	Flex port 3	If set, all packets can pass if match the TCP/UDP Port filter 3
16	Flex port 4	If set, all packets can pass if match the TCP/UDP Port filter 4
17	Flex port 5	If set, all packets can pass if match the TCP/UDP Port filter 5
18	Flex port 6	If set, all packets can pass if match the TCP/UDP Port filter 6
19	Flex port 7	If set, all packets can pass if match the TCP/UDP Port filter 7
20	Flex port 8	If set, all packets can pass if match the TCP/UDP Port filter 8
21	Flex port 9	If set, all packets can pass if match the TCP/UDP Port filter 9
22	Flex port 10	If set, all packets can pass if match the TCP/UDP Port filter 10
27:23	Reserved	
28	Flex TCO 0	If set, all packets can pass if match the Flex 128 TCO filter 0
29	Flex TCO 1	If set, all packets can pass if match the Flex 128 TCO filter 1
30	Flex TCO 2	If set, all packets can pass if match the Flex 128 TCO filter 2
31	Flex TCO 3	If set, all packets can pass if match the Flex 128 TCO filter 3

For extended packet reduction command, the following fields should also be programmed.

Table 10-18. Extended Packet Reduction Format

Bit #	Name	Description
0..1	Reserved	Used by the regular NC-SI commands
2	EtherType2 (AND)	If set, packets must also match the EtherType filter 2.
3	EtherType3 (AND)	If set, packets must also match the EtherType filter 3.
4..7	Reserved	
8..9	Reserved	Used by the regular NC-SI commands
10	EtherType2 (OR)	If set, packets can pass if it match the EtherType filter 2.
11	EtherType3 (OR)	If set, packets can pass if it match the EtherType filter 2.
12..31	Reserved	

The filtering is divided into 2 decisions:

Unicast Reduction Filter - Bit 3 and Extended Unicast Reduction Filter bits 0..2 work in an “AND” manner –Thus, they all must be true in order for a packet to pass (if any were set).

Unicast Reduction Filter Bits 5,7-31 and Extended Unicast Reduction Filter bits 8..10 work in an “OR” manner – Thus, at least one of them must be true for a packet to pass (if any were set).

See Section 10.4.4 for description of the decision filters.

10.3.1.9.2 Set Intel Packet Reduction Filters Response (Intel Command 0x04)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	Packet Reduction Index		

10.3.1.9.3 Set Unicast packet reduction command (Intel Command 0x04, Reduction Filter Index 0x00)

This command shall cause the Network Controller to filter packets that have passed due to the Unicast filter (MAC Address filters, as specified in the DMTF NC-SI). Note that Unicast filtering might be affected by other filters, as specified in the DMTF NC-SI.

The filtering of these packets will be done such that the Management Controller may add a logical condition that a packet must match, or it shall be discarded.

Note: Packets that might have been blocked may still pass due to other decision filters.

In order to disable Unicast packet reduction the Management Controller should set all reductions filters to 0b. Following such a setting the Network Controller shall forward to the BMC all packets that have passed the Unicast filters (MAC Address filtering) as specified in the DMTF NC-SI.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x00	Unicast Packet Reduction (MSB)	
24..25	Unicast Packet Reduction (LSB)			

See Table 10-17 for description of the Unicast Packet Reduction format.

10.3.1.9.4 Set Unicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x00)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x00		

10.3.1.9.5 Set Multicast packet reduction command (Intel Command 0x04, Reduction Filter Index 0x01)

This command shall cause the Network Controller to filter packets that have passed due to the Multicast filter (MAC Address filters, as specified in the DMTF NC-SI).

The filtering of these packets will be done such that the Management Controller may add a logical condition that a packet must match, or it shall be discarded.

Note: Packets that might have been blocked may still pass due to other decision filters.

In order to disable Multicast packet reduction the Management Controller should set all reductions filters to 0b. Following such a setting the Network Controller shall forward to the BMC all packets that have passed the Multicast filters (Global Multicast Filtering) as specified in the DMTF NC-SI.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x01	Multicast Packet Reduction (MSB)	
24..25	Multicast Packet Reduction (LSB)			

See Table 10-17 for description of the Multicast Packet Reduction format.

10.3.1.9.6 Set Multicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x01)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x01		

10.3.1.9.7 Set Broadcast packet reduction command (Intel Command 0x04, Reduction Filter Index 0x02)

This command shall cause the Network Controller to filter packets that have passed due to the Broadcast filter (MAC Address filters, as specified in the DMTF NC-SI).

The filtering of these packets will be done such that the Management Controller may add a logical condition that a packet must match, or it shall be discarded.

Note: Packets that might have been blocked may still pass due to other decision filters.

In order to disable Broadcast packet reduction the Management Controller should set all reductions filters to 0b. Following such a setting the Network Controller shall forward to the BMC all packets that have passed the Broadcast filters as specified in the DMTF NC-SI.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x02	Broadcast Packet Reduction (MSB)	
24..25	Broadcast Packet Reduction (LSB)			

See Table 10-17 for description of the Broadcast Packet Reduction format.

10.3.1.9.8 Set Broadcast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x02		

10.3.1.9.9 Set Unicast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x10)

In “Set Intel Reduction Filters” add another parameter “Unicast Extended Packet Reduction (Intel Command 0x04, Filter parameter 0x10)” such that the byte count is 0xE. The command shall have the following format:

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x10	Extended Unicast Reduction Filter MSB	..
24..27	..	Extended Unicast Reduction Filter LSB	Unicast Reduction Filter MSB	..
28..29	..	Unicast Reduction Filter LSB		

The command shall overwrite any previously stored value.

Note: See Table 10-17 and Table 10-18 for description of the Unicast Extended Packet Reduction format.

10.3.1.9.10 Set Unicast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x10)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x10		

10.3.1.9.11 Set Multicast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x11)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x11	Extended Multicast Reduction Filter MSB	..
24..27	..	Extended Multicast Reduction Filter LSB	Multicast Reduction Filter MSB	..
28..29	..	Multicast Reduction Filter LSB		

See Table 10-17 and Table 10-18 for description of the Multicast Extended Packet Reduction format.

The command shall overwrite any previously stored value.

10.3.1.9.12 Set Multicast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x11)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x11		

10.3.1.9.13 Set Broadcast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x12)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x12	Extended Broadcast Reduction Filter MSB	..
24..27	..	Extended Broadcast Reduction Filter LSB	Broadcast Reduction Filter MSB	..
28..29	..	Broadcast Reduction Filter LSB		

See Table 10-17 and Table 10-18 for description of the Broadcast Extended Packet Reduction format.

The command shall overwrite any previously stored value.

10.3.1.9.14 Set Broadcast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x12)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x12		

10.3.1.10 Get Intel Packet Reduction Filters Formats

10.3.1.10.1 Get Intel Packet Reduction Filters Command (Intel Command 0x05)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	Reduction Filter Index		

10.3.1.10.2 Get Intel Packet Reduction Filters Response (Intel Command 0x05)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..	0x05	Reduction Filter Index	Return Data	

See Table 10-17 and Table 10-18 for description of the Return data

10.3.1.10.3 Get Unicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x00)

This command shall cause the Network Controller to disable any packet reductions for Unicast address filtering.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x00		

10.3.1.10.4 Get Unicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x00)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x05	0x00	Unicast Packet Reduction (MSB)	
28..29	Unicast Packet Reduction (LSB)			

10.3.1.10.5 Get Multicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x01)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x01		

10.3.1.10.6 Get Multicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x01)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x05	0x00	Multicast Packet Reduction (MSB)	
28..29	Multicast Packet Reduction (LSB)			

10.3.1.10.7 Get Broadcast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x02		

10.3.1.10.8 Get Broadcast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x05	0x00	Broadcast Packet Reduction (MSB)	
28..29	Broadcast Packet Reduction (LSB)			

10.3.1.10.9 Get Unicast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x10)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x10		

10.3.1.10.10 Get Unicast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x10)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x05	0x10	Extended Unicast Reduction Filter MSB	..
28..31	..	Extended Unicast Reduction Filter LSB	Unicast Reduction Filter MSB	..
32..33	..	Unicast Reduction Filter LSB		

10.3.1.10.11 Get Multicast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x11)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x11		

10.3.1.10.12 Get Multicast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x11)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x05	0x11	Extended Multicast Reduction Filter MSB	..
28..31	..	Extended Multicast Reduction Filter LSB	Multicast Reduction Filter MSB	..
32..33	..	Multicast Reduction Filter LSB		

10.3.1.10.13 Get Broadcast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x12)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x12		

10.3.1.10.14 Get Broadcast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x12)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			

24..27	0x05	0x12	Extended Broadcast Reduction Filter MSB	..
28..31	..	Extended Broadcast Reduction Filter LSB	Broadcast Reduction Filter MSB	..
32..33	..	Broadcast Reduction Filter LSB		

10.3.1.11 System MAC Address

10.3.1.11.1 Get System MAC Address Command (Intel Command 0x06)

In order to support a system configuration that requires the NC to hold the MAC address for the BMC (i.e. – “Shared MAC Address” mode), the following command is provided to allow the BMC to query the NC for a valid MAC address.

The NC shall return the System MAC addresses. The BMC should use the returned MAC addressing as a shared MAC Address by setting it using the “Set MAC Address” command as defined in NC-SI 1.0.

It is also recommended that the BMC uses the Packet Reduction & Manageability to Host command to set the proper filtering method.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x06			

10.3.1.11.2 Get System MAC Address Response (Intel Command 0x06)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..27	0x06	MAC Address		
28..30	MAC Address			

The MAC address is returned in Network Order.

10.3.1.12 Set Intel Management Control Formats

10.3.1.12.1 Set Intel Management Control Command (Intel Command 0x20)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x20	0x00	Intel Management Control	

Where:

“Intel Management Control” is as follows.

Table 10-19. Intel Management Control

Bit #	Default value	Description
0	0b	Enable Critical Session Mode (a.k.a. “Keep PHY Link Up”, “Veto Bit”) 0b - Disabled 1b - Enabled When Critical Session Mode is enabled, the following behaviors are disabled: The PHY is not reset on PE_RST# and PCIe resets (in-band and link drop). Other reset events are not affected - Internal_Power_On_Reset reset, Device Disable, Force TCO, and PHY reset by SW. The PHY does not change its power state. As a result link speed does not change. The device does not initiate configuration of the PHY to avoid losing link.
1...7	0x0	Reserved

10.3.1.12.2 Set Intel Management Control Response (Intel Command 0x20)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x20	0x00		

10.3.1.13 Get Intel Management Control Formats

10.3.1.13.1 Get Intel Management Control Command (Intel Command 0x21)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x20	0x00		

10.3.1.13.2 Get Intel Management Control Response (Intel Command 0x21)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..26	0x21	0x00	Intel Management Control	

"Intel Management Control" is as described in "Set Intel Management Control" (Section 10.3.1.12).

10.3.1.14 TCO Reset

This command will cause the Network Controller to perform TCO Reset, if Force TCO reset is enabled in the EEPROM.

If the BMC has detected that the OS is hung and has blocked the RX/TX path The Force TCO reset will clear the data-path (RX/TX) of the Network Controller to enable the BMC to transmit/receive packets through the Network Controller.

When this command is issued to a channel in a package, it applies only to the specific channel.

After successfully performing the command the Network Controller will consider Force TCO command as an indication that the OS is hung and will clear the DRV_LOAD flag (disable the driver).

10.3.1.14.1 Perform Intel TCO Reset Command (Intel Command 0x22)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x22			

10.3.1.14.2 Perform Intel TCO Reset Response (Intel Command 0x22)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..26	0x22			

10.3.1.15 Checksum offloading

This command will enable the checksum offloading filters in the Network Controller.

When enabled, these filters will block any packets that did not pass IP, UDP & TCP checksum from being forwarded to the Management Controller.

10.3.1.15.1 Enable Checksum Offloading Command (Intel Command 0x23)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x23			

10.3.1.15.2 Enable Checksum Offloading Response (Intel Command 0x23)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..26	0x23			

10.3.1.15.3 Disable Checksum Offloading Command (Intel Command 0x24)

This command shall cause the Network Controller to stop verifying the IP/UDP/TCP checksum.

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x24			

10.3.1.15.4 Disable Checksum Offloading Response (Intel Command 0x24)

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..26	0x24			

10.3.2 SMBus Programming

This section describes the SMBus transactions supported in APT mode.

10.3.2.1 Write SMBus Transactions (BMC → Intel® 82576 GbE Controller)

The table below summarizes the different SMB write transactions supported by the 82576.

Table 10-20. SMB Write Transactions

TCO Command	Transaction	Command		Fragmentation	Section
Transmit Packet	Block Write	First: Middle: Last:	0x84 0x04 0x44	Multiple	10.3.2.1.1
Transmit Packet	Block Write	Single:	0xC4	Single	10.3.2.1.1
Receive Enable	Block Write	Single:	0xCA	Single	10.3.2.1.3
Management Control	Block Write	Single:	0xC1	Single	10.3.2.1.5
Update MNG RCV filter parameters	Block Write	Single:	0xCC	Single	10.3.2.1.6
Force TCO	Block Write	Single:	0xCF	Single	10.3.2.1.4
Request Status	Block Write	Single:	0xDD	Single	10.3.2.1.2

10.3.2.1.1 Transmit Packet Command

The Transmit Packet command behavior is detailed in section 3.2.1.1.4.

The Transmit Packet fragments will have the following format.

Function	Command	Byte Count	Data 1	...	Data N
Transmit first fragment	0x84	N	Packet data MSB	...	Packet data LSB
Transmit middle fragment	0x04				
Transmit last fragment	0x44				
Transmit single fragment	0xC4				

The payload length is limited to the maximum payload length set in the EEPROM.

If the overall packet length is bigger than 1536 bytes, the packet is silently discarded by the 82576.

10.3.2.1.2 Request Status Command

The TCO controller can initiate a request to read the 82576 manageability status by sending this command.

When it receives this command, the 82576 initiates a notification to the external controller (when it is ready with the status), and then the external controller will be able to read the status, by issuing a read status command (see section 10.3.2.2.3).

Request Status Command format:

Function	Command	Byte Count	Data 1
Request status	0xDD	1	0

10.3.2.1.3 Receive Enable Command

The Receive Enable command is a single fragment command that is used to configure the 82576.

This command has two formats: short, 1-byte legacy format (providing backward compatibility with previous components) and long, 14-byte advanced format (allowing greater configuration capabilities).

Note: If the receive enable command is short and thus does not include all the parameters, then the parameters will be taken from most recent previous configuration (either the most recent long Receive-Enable command in which the particular value was set, or the EEPROM if there was no such previous long Receive-Enable command).

Func.	Cmd	Byte Count	Data 1	Data 2	...	Data 7	Data 8	...	Data 11	Data 12	Data 13	Data 14
Legacy receive enable	0xCA	1	Receive control byte	-	...	-	-	...	-	-	-	-
Advanced receive enable		14 0x0E		MAC addr. MSB		MAC addr. LSB	IP addr. MSB		IP addr. LSB	BMC SMBus addr.	Interf. data byte	Alert value byte

Where:

- Receive control byte (data byte 1):

Field	Bit(s)	Description
RCV_EN	0	Receive TCO enable. 0b = Disable Receive TCO packets 1b = Enable Receive TCO packets. Setting this bit will enable all manageability receive filtering operation. The enable of the specific filtering is done through loading the EPROM, or through special configuration command (see Section 10.3.2.1.6 and Section 6.9). **Note:** When this bit is cleared all receive TCO functionality is disabled, not just the packets that are directed to the BMC (also auto ARP packets).
RCV_ALL	1	Receive All enable. 0b = Disable receiving all packets 1b = Enable receiving all packets Will forward all LAN packets received over the wire that passed L2 filtering to the BMC. This flag will have no affect if bit 0 (Enable TCO packets) is disabled
EN_STA	2	Enable Status reporting. 0b = Disable status reporting 1b = Enable status reporting
EN_ARP_RES	3	Enable ARP response 0b = Disable the 82576 ARP response. The 82576 will treat ARP packets as any other packet, such as a packet is forwarded to BMC if it passed other (non-ARP) filtering. 1b = Enable the 82576 ARP response The 82576 will automatically respond to all received ARP requests which match its IP address. Note that setting this bit doesn't change the Rx filtering settings. Appropriate Rx filtering to enable ARP request packets to reach manageability unit should be set by the BMC or by the EEPROM. The BMC IP address is provided as part of the Receive Enable message (bytes 8-11). If short version of the command is used the 82576 will use IP address configured in the most recent long version of the command in which the EN_ARP_RES bit was set. If no such previous long command exists, then the 82576 will use IP address configured in the EEPROM as ARP Response IPv4 Address in PT LAN configuration structure. If CBDM bit is set the 82576 will use BMC dedicated MAC address in ARP response packets. If the CBDM bit is not set, BMC will use the host MAC address. Setting this bit requires appropriate assertion of bits RCV_EN and RCV_ALL. Otherwise, the command aborts with no processing.
NM	4-5	Notification Method. Define the notification method the Intel® 82576 GbE Controller will use. 00b = SMB Alert 01b = Asynchronous Notify 10b = Direct Receive 11b = Not Supported. **Note:** In dual SMB address mode, both SMB addresses must be configured to the same notification method. **Note:** In FML mode the notification method must be SMB Alert (00b).
Reserved	6	Reserved, must be 1b.

CBDM	7	Configure BMC dedicated MAC Address. **Note:** This bit should be zero when the RCV_EN bit (bit 0) is not set. 0b = The 82576 will share MAC address for MNG traffic with host MAC address which is specified in EEPROM words 0x0-0x2. 1b = The 82576 will use the BMC dedicated MAC address as filter for incoming receive packets and as sender address in ARP response packets. The BMC MAC address is set in bytes 2-7 in this command. If short version of the command is used, the 82576 will use MAC address configured in the most recent long version of the command in which the CBDM bit was set. If no such previous long command exists, then the 82576 will use MAC address configured in the EEPROM as MAC address 3 (MMAL/H3) in PT LAN configuration structure. When Dedicated MAC address feature is activated, the 82576 will use the following registers to filter in all the traffic addressed to the BMC MAC. BMC should not modify these registers: Manageability Decision Filter – MDEF7 (and corresponding bit 7 in Management Control To Host Register – MANC2H) Manageability MAC Address Low – MMAL3 Manageability MAC Address High – MMAH3 (and corresponding bit 3 of Manageability Filters Valid – MFVAL).

- MNG MAC Address (data bytes 2-7)

Ignored if CBDM bit is not set. This MAC address will be used for configuration of the dedicated MAC address. In addition, it will be used in the ARP response packet, when EN_ARP_RES bit is set. This MAC address will continue to be used when CBDM bit is set in subsequent short versions of this command.

- MNG IP address (data bytes 8-11)

Ignored if EN_ARP_RES bit is not set. This IP address will be used to filter ARP request packets. This IP address will continue to be used when EN_ARP_RES is set in subsequent short versions of this command.

- Asynchronous Notification SMB address (data byte 12)

This address will be used for the Asynchronous Notification SMB transaction and for Direct Receive.

- Interface Data (Data byte 13)

Interface data byte to be used in Asynchronous Notification.

- Alert data (Data byte 14).

Alert Value data byte to be used in the Asynchronous Notification.

10.3.2.1.4 Force TCO Command

This command will cause the 82576 to perform TCO Reset, if Force TCO reset is enabled in the EEPROM (see Section 6.5.9). The Force TCO reset will clear the data-path (RX/TX) of the 82576 to enable the BMC to transmit/receive packets through the 82576.

Note: In single address mode both ports will be reset when the command is issued. In dual address Force TCO reset will be asserted only to the port related to the SMB address the command was issued to.

The 82576 will consider Force TCO command as an indication that the OS is hung and will clear the DRV_LOAD flag.

Force TCO Reset Command format:

Function	Command	Byte Count	Data 1
Force TCO reset	0xCF	1	TCO mode

Where:

TCO Mode:

Field	Bit(s)	Description
DO_TCO_RST	0	Do TCO reset. 0b = Do Nothing 1b = Perform TCO Reset
Reserved	1-7	Reserved. (Set to 0x00)

10.3.2.1.5 Management Control

This command is used to set generic manageability parameters. The parameters list is shown in the table below. The command is C1h, which states that it is a management control command. The first data byte is the parameter number and the data after-words (length and content) are parameter specific as shown in the table.

Note: If in the update configuration, the parameter that the BMC sets is not supported by the 82576, the 82576 will not NACK the transaction. After the transaction ends, the 82576 will discard the data and will assert a transaction abort status (see Section 3.2.1.1.6).

This is the format of the Management control command:

Function	Command	Byte Count	Data 1	Data 2	...	Data N
Management control	0xC1	N	Parameter number	Parameter dependent		

The table below shows the different parameters and their content:

Parameter	Parameter Number	Parameter Data
Keep PHY Link Up	0x00	A single byte parameter: Data 2 - Bit 0 Set to indicate that the PHY link for this port should be kept up. Sets the keep_PHY_link_up bit. When cleared, clears the keep_PHY_link_up bit. See Section 8.21.4. Bit [7:1] Reserved See Section 4.2.3 for details of this bit behavior.

10.3.2.1.6 Update MNG RCV Filter Parameters

This command is used to set the manageability receive filters parameters. The parameters list is shown in the table below. The command is 0xCC, which states that it is a parameter update. The first data byte is the parameter number and the data after-words (length and content) are parameter specific as shown in the table.

Note: If in the update configuration, the parameter that the BMC sets is not supported by the 82576, the 82576 will not NACK the transaction. After the transaction ends, the 82576 will discard the data and will assert a transaction abort status (see Section 3.2.1.1.6).

Detailed description of receive filtering capabilities and configuration is in Section 10.4

This is the format of the Update MNG RCV filter parameters command:

Function	Command	Byte Count	Data 1	Data 2	...	Data N
Update MNG RCV filter parameters	0xCC	N	Parameter number	Parameter dependent		

The table below shows the different parameters and their contents.

Parameter	#	Parameter Data
Filters Enable	0x1	Defines generic filters configuration. The structure of this parameter is 4 bytes as the MANC register shown in EEPROM words 2Fh-30h of the PT LAN configuration structure (see Section 6.7.17 and Section 6.7.18). Note the general filter enable is in the receive enable command. Which enable receive filtering. This parameter specifies which filters should be enabled. ARP filtering and dedicated MAC address can also be enabled through the receive enable command (see Section 10.3.2.1.3).
MNG2HOST configuration	0xA	This parameter defines which of the packets types identified as manageability packets in the receive path will be directed to the HOST memory too. Data 2:5: MNG2H register bits (see Section 8.21.6) - Data 2 is the MSB
Fail-Over configuration	0xB	Fail-Over register configuration – The bytes of this parameter will be loaded into the Fail-over configuration register. Data 2:5– Fail Over configuration register (Data 2 is the MSB). See Table 10-3 for detailed description of the Fail-Over configuration register.
Flex filter 0 enable MASK and length	0x10	Flex filter 0 mask. Data 2:17 – MASK. Bit 0 in data 2 is the first bit of the MASK Data 18:19 – Reserved. Should be zero. Data 20 – Flexible Filter length. Must be greater than or equal one.
Flex filter 0 data	0x11	Data 2 – Group of Flex filter's bytes: 0x0 = bytes 0-29 0x1 = bytes 30-59 0x2 = bytes 60-89 0x3 = bytes 90-119 0x4 = bytes 120-127 Data 3:32 – Flex filter data bytes. Data 3 is LSB. Group's length is not mandatory 30 bytes; it may vary according to filter's length and must NOT be padded by zeros.
Flex filter 1 enable MASK and length	0x20	Same as parameter 0x10 but for filter 1.
Flex filter 1 data	0x21	Same as parameter 0x11 but for filter 1
Flex filter 2 enable MASK and length	0x30	Same as parameter 0x10 but for filter 2.
Flex filter 2 data	0x31	Same as parameter 0x11 but for filter 2.
Flex filter 3 enable MASK and length	0x40	Same as parameter 0x10 but for filter 3.
Flex filter 3 data	0x41	Same as parameter 0x11 but for filter 3.
Filters Valid	0x60	4 bytes to determine which of the Intel® 82576 GbE Controller filter registers contain valid data. Loaded into the MFVAL register. Should be updated after the contents of a filter register are updated. Data 2 – MSB of MFVAL ... Data 5 – LSB of MFVAL

Decision Filters	0x61	5 bytes to load the manageability decision filters (MDEF) Data 2 – Decision Filter number Data 3 – MSB of MDEF register for this decision filter ... Data 6 – LSB of MDEF register for this decision filter
VLAN Filters	0x62	3 bytes to load the VLAN tag filters (MAVTV) Data 2 – VLAN Filter number Data 3 – MSB of VLAN Filter Data 4 – LSB of VLAN Filter
Flex Ports Filters	0x63	3 bytes to load the manageability flex port filters (MFUTP). Data 2 – Flex Port Filter number Data 3 – MSB of flex port filter Data 4 - LSB of flex port filter
IPv4 Filters	0x64	5 bytes to load the IPv4 address filter (MIPAF, DW 15:12) Data 2 – IPv4 address filter number (0-3) Data 3 – MSB of IPv4 address filter ... Data 6 – LSB of IPv4 address filter
IPv6 Filters	0x65	17 bytes to load IPv6 address filter (MIPAF) Data 2 – IPv6 address filter number (0-3) Data 3 – MSB of IPv6 address filter ... Data 18 – LSB of IPv6 address filter
MAC Filters	0x66	7 bytes to load MAC address filters (MMAL, MMAH) Data 2 – MAC address filters pair number (0-3) Data 3 – MSB of MAC address ... Data 8 – LSB of MAC address
Ethertype Filters	0x67	5 bytes to load Ethertype filters (METF) Data 2 - METF filter index (valid values are 0, 1, 2, 3) Data 3 - MSB of METF ... Data 6 - LSB of METF
Extended Decision Filter	0x68	9 bytes to load the extended decision filters (MDEF_EXT & MDEF) Data 2 - MDEF filter index (valid values are 0..6) Data 3 - MSB of MDEF_EXT (DecisionFilter1) .... Data 6 - LSB of MDEF_EXT (DecisionFilter1) Data 7 - MSB of MDEF (DecisionFilter0) .... Data 10 - LSB of MDEF (DecisionFilter0) The command shall overwrite any previously stored value.

10.3.2.2 Read SMBus Transactions (Intel® 82576 GbE Controller to BMC)

The table below summarizes the different SMB read transactions supported by the 82576. The Receive TCO Packet commands are not used in NC-SI+SMB modes. All the read transactions are compatible with SMBus Read Block Protocol format.

TCO Command	Transaction	Command	Op-Code		Fragmentation	Section
Receive TCO Packet	Block Read	0xC0 or 0xD0	First: Middle: Last[1]	0x90 0x10 0x50	Multiple	10.3.2.2.1
Read configuration	Block Read	0xC6 + 0xD2	Single:	0xD2	Single	10.3.2.2.5

Read Receive Enable configuration	Block Read	0xDA	Single:	0xDA	Single	10.3.2.2.7
Read the 82576 Status	Block Read	0xC0 or 0xD0 or 0xDE	Single:	0xDD	Single	10.3.2.2.3
Read Management parameters	Block Read	0xD1	Single:	0xD1	Single	10.3.2.2.5
Read MNG RCV filter parameters	Block Read	0xCD	Single:	0xCD	Single	10.3.2.2.6
Get system MAC address	Block Read	0xD4	Single	0xD4	Single	10.3.2.2.4

1. Last fragment of the receive TCO packet is the packet status.

Note: The 82576 will respond to one of the commands 0xC0/0xD0 within the time defined in the SMBus notification timeout and flags word in the EEPROM (See Section 6.8.3.)

0xC0/0xD0 commands are used for more than one payload. If the BMC issues these read commands, and the 82576 has no pending data to transfer, it will always return as default opcode 0xDD with the 82576 status, and will not NACK the transaction.

If an SMBus quick read command is received, it is handled as a Read the 82576 Status command (See Section 10.3.2.2.3 for details).

10.3.2.2.1 Receive TCO LAN Packet Transaction

The BMC will use this command to read the packet received on the LAN and its status. When the 82576 has a packet to deliver to the BMC, it asserts the SMB notification, for the BMC to read the data (or direct receive). Upon receiving notification of the arrival of LAN receive packet, the BMC should begin issuing a Receive TCO packet command using the block read protocol. The packet can be delivered in more than one SMB fragment (at least two - one for the packet, and the other one for the status), and the BMC should follow the 'F' and 'L' bit.

The op-code can have these values:

- 0x90 - First Fragment
- 0x10 - Middle Fragment.
- 0x50 - Packet status (last fragment) as described below in Section 10.3.2.2.2.

If the external BMC will not finish reading the whole packet within a timeout period since the last fragment was read, the packet will be silently discarded. The timeout period is set according to the SMBus notification timeout EEPROM parameter. See Section 6.8.3.

Function	Command
Receive TCO packet	0xC0 or 0xD0

Data returned from the 82576:

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data N

Receive TCO First Fragment	N	90	Packet Data Byte	...	Packet Data Byte
Receive TCO Middle Fragment		10			
Receive TCO Last Fragment (TCO status, see Section 10.3.2.2.2)		50			

10.3.2.2.2 Receive TCO LAN Status Payload Transaction

This transaction is the last transaction that the 82576 will issue when a packet that was received from the LAN is transferred to the BMC. The transaction will contain the status of the received packet.

This is the format of the status transaction:

Function	Byte Count	Data 1 (Op-Code)	Data 2 – Data 17 (Status data)
Receive TCO long status	17 (0x11)	0x50	See below

The status is 16 bytes where byte 0 (bits 0-7) is set in Data 2 of the status and byte 15 is set in Data 17 of the status.

The table below describes the content of the status data

Table 10-21. TCO LAN Packet Status Data

Name	Bits	Description
Packet Length	13:0	Packet length including CRC, only 14 LSB bits.
Packet status	35:14	See Table 10-22
Reserved	42:36	Reserved
Error	47:43	See Table 10-23
VLAN	63:48	The two bytes of the VLAN header tag.
Reserved	67:64	Reserved
Packet type	80:68	See Table 10-25
Reserved	84:81	Reserved
MNG status	127:85	See Table 10-26. This field should be ignored if Receive TCO is not enabled,

The meaning of the bits inside of each field can be found in Section 7.1.5.

Table 10-22. Packet Status Info

Field	Bit(s)	Description
LAN#	21	Indicates the source port of the packet
Reserved	20	Reserved
VP	19	VLAN Stripped –insertion of VLAN TAG is needed.
VEXT	18	Additional VLAN present in packet
Reserved	17:15	Reserved
Reserved	14	Reserved

Table 10-22. Packet Status Info

Reserved	13:12	Reserved
CRC stripped	11	Insertion of CRC is needed.
Reserved	10:6	Reserved
UDPV	5	UDP checksum valid
Reserved	4:3	Reserved
IPCS	2	Ipv4 Checksum Calculated on packet
L4I	1	L4 (TCP/UDP) Checksum calculated on packet
UDPCS	0	UDP checksum calculated on packet

Table 10-23. Error Status Info

Field	Bit(s)	Description
RXE	4	RX Data Error
IPE	3	Ipv4 Checksum Error
L4E	2	L4 (TCP/UDP) Checksum Error
SECE	1:0	Security Error - See Table 10-24

Table 10-24. Security errors

Code	Error Type
00b	No error
01b	No SA match
10b	Replay error
11b	Bad signature

Table 10-25. Packet type

Bit Index	Bit 11 = 0b	Bit 11 = 1b (L2 packet)
12	VLAN packet indication	
11	Packet matched one of the ETQF filters.	
10	Reserved	Reserved
9	IPSec AH – IPSec encapsulation	Reserved
8	IPSec ESP – IPSec encapsulation	
7	NFS - NFS header present	
6	SCTP - SCTP header present	
5	UDP - UDP header present	
4	TCP - TCP header present	
3	IPV6E- IPv6 Header includes extensions	
2	IPV6 - IPv6 header present	EtherType - ETQF register index that matches the packet. Special types might be defined for 1588, 802.1X, or any other requested type.
1	IPV4E - IPv4 Header includes extensions	
0	IPV4 - IPv4 header present	

Table 10-26. MNG Status

Name	Bits	Description
Decision Filter match	42:35	Set when there is a match to one of the Decision filters
IPv4/IPv6 match	34	Set when there is an IPv4 match or IPv6 match. This bit is valid only if the bit 30 (IP match bit) or bit 4 (ARP match bit) are set.
IP address match	33	Set when there is a match to any of the IP address filters
IP address Index	32:31	Set when there is a match to the IP filter number. (IPv4 or IPv6)
Flex TCO filter match	30	
Flex TCO Filter index	29:27	
L4 port match	26	Set when there is a match to any of the UDP / TCP port filters
L4 port Filter Index	25:19	Indicate the flex filter number
Unicast Address match	18	Set when there is a match to any of the 4 Unicast MAC addresses.
Unicast Address Index	17:15	Indicates which of the 4 Unicast MAC addresses match the packet. Valid only if the Unicast Address match is set.
MNG VLAN Address Match	14	Set when the MNG packet matches one of the MNG VLAN filters
Pass MNG VLAN Filter Index	13:11	
Reserved	10:8	Reserved
Pass ARP req / ARP resp	7	Set when the MNG packet is an ARP response/request packet
Pass MNG neighbor	6	Set when the MNG packet is a neighbor discovery packet.
Pass MNG broadcast	5	Set when the MNG packet is a broadcast packet
Pass RMCP 0x0298	4	Set when the UDP/TCP port of the MNG packet is 0x298
Pass RMCP 0x026F	3	Set when the UDP/TCP port of the MNG packet is 0x26F
Manageability Ethertype filter passed	2	Indicates that one of the METF filters matched
manageability Ethertype filter index	1:0	Indicates which of the METF filters matched

10.3.2.2.3 Read Status Command

The BMC can read the 82576 status. the 82576 will assert an alert prior to the BMC reading the status bytes. There can be two reasons for the 82576 to send status to the BMC:

1. The external BMC asserts a request for reading the 82576 status (Section 3.2.1.1.2).
2. The 82576 detects a change in one of the “status Data 1" bits or bits 4 of “status Data 2", and was set to send status to the BMC, on status change, in the receive enable command (Section 3.2.1.1.2).

Note that commands C0h/D0h are for backward compatibility. 0xD0/0xC0 can be used for other payloads the 82576 will define in the op-code, which payload this transaction will be. When 0xDE command is set, the 82576 will always return opcode 0xDD with the 82576 status. The BMC will read the event causing the notification, using the read status command shown below:

Function	Command
Read status	0xC0 or 0xD0 or 0xDE

Function	Byte Count	Data 1 (Op-Code)	Data 2 (Status data 1)	Data 3 (Status data 2)
Receive TCO partial status	3	0xDD	See below	

The table below shows status data byte 1:

Bit	Name	Description
7	LAN Port	0b = Status came from LAN port 0. 1b = Status came from LAN port 1
6	TCO command aborted	0b = A TCO command abort event has not occurred since the last Read Status cycle. 1b = A TCO command abort event has occurred since the last Read Status cycle. See Section 3.2.1.1.6 for command abort flow.
5	Link Status indication	0b = LAN link down 1b = LAN link is up See note 1
4	PHY Link Forced Up	Contains the value of the MANC.keep_PHY_link_up bit. When set, indicates that the PHY link is configured to keep link up. See Section 8.21.4.
3	Initialization indication	0b = An EEPROM reload event has not occurred since the last Read Status cycle. 1b = An EEPROM reload event has occurred since the last Read Status cycle. See note 2.
2	Reserved	Reserved as 0b
1:0	Power state	00b = Dr state 01b = D0u state 10b = D0 state 11b = D3 state See note 3.

Notes:

1. When the 82576 is working in teaming mode, and presented as one SMB device, the link indication will be 0b only when both links (on both ports) are down. If one of the LANs is disabled, its link is considered to be down.
2. This indication will be asserted when the the 82576 manageability block will reload EEPROM and its internal data-base will be updated to EEPROM default values. This is an indication that the external BMC should re-configure the 82576, if other values beside the EPROM default should be configured.
3. In Single address mode the 82576 will report the highest Power-state modes in both devices. The “D” state is marked in this order: D0, D0u, Dr, and D3.

Status data byte 2 is used for the BMC for an indication whether the LAN driver is alive and running.

The driver valid indication is a bit that is set by the driver when it is coming up, and cleared when it goes down to Dx state or cleared by the HW on PCI reset.

Bits 2 and 1 indicate that the LAN driver is not stuck. Bit 2 indicates whether the interrupt line of the LAN function is asserted, and bit 1 indicates whether driver between the last Read Status Cycle dealt the interrupt line.

The table below shows status data byte 2:

Bit	Name	Description
7	Reserved	Reserved
6	Reserved	Reserved
5	Reserved	Reserved
4	Reserved	Reserved
3	Driver Valid indication	0b = LAN driver is not alive 1b = LAN driver is alive
2	Interrupt pending indication	0b = LAN interrupt is not asserted. 1b = LAN interrupt is asserted.
1	ICR register read/write	0b = ICR register was not read since the last Read Status Cycle. 1b = ICR register was read since the last Read Status cycle. Reading the ICR means that the driver has dealt with the interrupt that was asserted
0	Reserved	Reserved

Note: When the 82576 is in teaming mode, the above bits represent both cores:

- The driver alive indication is set if one of the driver is alive.
- The LAN interrupt consider to be asserted if one of the interrupt lines is asserted.
- The ICR is considered to read if one of the ICRs was read (LAN0 or LAN1).

The table below shows the possible values of bits 2, 1 and what the BMC can assume according to that:

Previous	Current	
Don’t care	00b	Interrupt is not pending – O.K
00b	01b	New interrupt is asserted – O.K
10b	01b	New interrupt is asserted – O.K
11b	01b	Interrupt is waiting for reading – O.K
01b	01b	Interrupt is waiting for reading by driver more than one read status Cycle – Not OK (possible driver hang state).
Don’t Care	11b	Previous interrupt was read and current interrupt is pending – O.K
Don’t Care	10b	Interrupt is not pending – O.K

Note: The BMC reads should consider the time it takes for the driver to deal with the interrupt (few micro-seconds), too frequent reads will give false indications.

10.3.2.2.4 Get System MAC Address

The Get System MAC Address will return system MAC Address over the SMBus. This command is a single fragment Read Block transaction, which will return the following database:

The Get System MAC Address returns the system MAC Address (RAL0, RAH0).

Get System MAC Address format:

Function	Command
Get system MAC address	0xD4

Data returned from the 82576:

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data 7
Get system MAC address	7	0xD4	MAC address MSB	...	MAC address LSB

10.3.2.2.5 Read Management Parameters

In order to read the management parameters the BMC should execute two SMB transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that reads the parameter.

This is the block write transaction:

Function	Command	Byte Count	Data 1
Management control Request	0xC1	1	Parameter number

Following the block write the BMC should issue a block read that will read the parameter that was set in the block write command:

Function	Command
Read management parameter	0xD1

Data returned from the 82576:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read management parameter	N	0xD1	Parameter number	Parameter dependent		

The returned data is as follow:

Parameter	#	Parameter Data
Keep PHY Link Up	0x00	A single byte parameter: Data 2 - Bit 0 Set to indicate that the PHY link for this port should be kept up. Sets the keep_PHY_link_up bit. When cleared, clears the keep_PHY_link_up bit. See Section 8.21.4. Bit [7:1] Reserved See Section 4.2.3 for details of this bit behavior.
Wrong parameter request	0xFE	Returned by the 82576 only. This parameter is returned on read transaction, if in the previous read command the BMC sets a parameter that is not supported by the 82576.
The 82576 is not ready	0xFF	Returned by the 82576 only, on read parameters command when the data that should have been read is not ready. This parameter has no data. The BMC should retry the read transaction.

Note: It might be that the parameter that is returned is not the parameter requested by the BMC. The BMC should verify the parameter number (default parameter to be returned is 1h).

It is BMC's responsibility to follow the procedure defined above. If the BMC sends a block read command (as described above) which is not preceded by a block write command with bytecount=1, the 82576 will set Parameter Number in the read block transaction to be 0xFE.

10.3.2.2.6 Read MNG RCV Filter Parameters

In order to read the MNG RCV filter parameters, the BMC should execute two SMB transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that reads the parameter.

This is the block write transaction:

Function	Command	Byte Count	Data 1	Data 2
Update MNG RCV filter parameters	0xCC	1 or 2	Parameter number	Parameter data

The table below shows the different parameters and their contents**:**

Parameter	#	Parameter Data
Filters Enable	0x1	None
MNG2HOST configuration	0xA	None
Fail-Over configuration	0xB	None
Flex filter 0 enable MASK and length	0x10	None
Flex filter 0 data	0x11	Data 2 – Group of Flex filter's bytes: 0x0 = bytes 0-29 0x1 = bytes 30-59 0x2 = bytes 60-89 0x3 = bytes 90-119 0x4 = bytes 120-127
Flex filter 1 enable MASK and length	0x20	None
Flex filter 1 data	0x21	Same as parameter 0x11 but for filter 1.
Flex filter 2 enable MASK and length	0x30	None
Flex filter 2 data	0x31	Same as parameter 0x11 but for filter 2.
Flex filter 3 enable MASK and length	0x40	None
Flex filter 3 data9	0x41	Same as parameter 0x11 but for filter 3.
Filters Valid	0x60	None
Decision Filters	0x61	1 byte to define the accessed manageability decision filter (MDEF) Data 2 – Decision Filter number
VLAN Filters	0x62	1 byte to define the accessed VLAN tag filter (MAVTV) Data 2 – VLAN Filter number
Flex Ports Filters	0x63	1 byte to define the accessed manageability flex port filter (MFUTP). Data 2 – Flex Port Filter number
IPv4 Filter	0x64	1 byte to define the accessed IPv4 address filter (MIPAF) Data 2 – IPv4 address filter number
IPv6 Filters	0x65	1 byte to define the accessed IPv6 address filter (MIPAF) Data 2 – IPv6 address filter number

MAC Filters	0x66	1 byte to define the accessed MAC address filters pair (MMAL, MMAH) Data 2 – MAC address filters pair number (0-3)
Wrong parameter request	0xFE	Returned by the 82576 only. This parameter is returned on read transaction, if in the previous read command the BMC sets a parameter that is not supported by the 82576.
The 82576 is not ready	0xFF	Returned by the 82576 only, on read parameters command when the data that should have been read is not ready. This parameter has no data.

Following the block write the BMC should issue a block read that will read the parameter that was set in the block write command:

Function	Command
Request MNG RCV filter parameters	0xCD

Data returned from the 82576:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read MNG RCV filter parameters	N	0xCD	Parameter number	Parameter dependent		

The returned data is in the same format of the “update” command. 1

Note: If the parameter that is returned is not the parameter requested by the BMC, the BMC should verify the parameter number (default parameter to be returned is 1h).

If the parameter number is 0xFF, it means that the data that the 82576 should supply is not ready yet. The BMC should retry the read transaction.

It is BMC’s responsibility to follow the procedure defined above. If the BMC sends a block read command (as described above) which is not preceded by a block write command, the 82576 will set Parameter Number in the read block transaction to be 0xFE.

10.3.2.2.7 Read Receive Enable Configuration

The BMC uses this command to read the receive configuration data. This data can be configured in receive enable command or through EPROM loading upon power-up.

Read Receive Enable Configuration command format (SMBus Read Block Protocol):

Function	Command
Read receive enable	0xDA

Data returned from the 82576.

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data 8	Data 9	...	Data 12	Data 13	Data 14	Data 15

Read receive enable	15 (0x0F)	0xDA	Receive control byte	MAC address MSB	...	MAC address LSB	IP address MSB	...	IP address LSB	BMC SMBus address	Interface data byte	Alert value byte

The detailed description of each field is specified in the receive enable command description in Section 10.3.2.1.3.

10.3.2.3 SMB ARP Transactions

Note: All SMB-ARP transactions include PEC byte.

10.3.2.3.1 Prepare to ARP

This command will clear the Address Resolved flag (set to false). It will not affect the status or validity of the dynamic SMBus Address. It is used to inform all devices that the ARP Master is starting the ARP process:

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0001	0	[Data dependent value]	0	

10.3.2.3.2 Reset Device (General)

This command will clear the Address Resolved flag (set to false). It will not affect the status or validity of the dynamic SMBus Address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0010	0	[Data dependent value]	0	

10.3.2.3.3 Reset Device (Directed)

The Command field is NACK-ed if the bits 7 through 1 do not match the current the 82576 SMBus address.

It will clear the Address Resolved flag (set to false). It will not affect the status or validity of the dynamic SMBus Address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	Targeted slave address | 0	0	[Data dependent value]	0	

10.3.2.3.4 Assign Address

This command assigns the 82576 SMBus address. The address and command bytes are always acknowledged.

The transaction will be aborted immediately (NACK-ed in simple English) if any of the UDID bytes differ from the 82576 UDID bytes as defined in Section 3.2.1.1.8. If successful, the MNG will update the SMBus address internally. This command will also set the Address Resolved flag to true.

1	7	1	1	8	1	8	1	
S	Slave Address	Wr	A	Command	A	Byte Count	A	•••
	1100 001	0	0	0000 0100	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data-1	A	Data-2	A	Data-3	A	Data-4	A	•••
UDID byte 15 (MSB)	0	UDID byte 14	0	UDID byte 13	0	UDID byte 12	0	

8	1	8	1	8	1	8	1	
Data-5	A	Data-6	A	Data-7	A	Data-8	A	•••
UDID byte 11	0	UDID byte 10	0	UDID byte 9	0	UDID byte 8	0	

8	1	8	1	8	1	
Data-9	A	Data-10	A	Data-11	A	•••
UDID byte 7	0	UDID byte 6	0	UDID byte 5	0	

8	1	8	1	8	1	8	1	
Data-12	A	Data-13	A	Data-14	A	Data-15	A	•••
UDID byte 4	0	UDID byte 3	0	UDID byte 2	0	UDID byte 1	0	

8	1	8	1	8	1	1
Data-16	A	Data-17	A	PEC	A	P
UDID byte 0 (LSB)	0	Assigned Address	0	[Data dependent value]	0	

10.3.2.3.5 Get UDID (General and Directed)

The Get UDID command depends on whether this is a directed or general command.

The General Get UDID SMBus transaction will support a constant command value of 0x03.

The Directed Get UDID SMBus transaction will support a dynamic command value equal to the dynamic SMBus address with the LSB bit set.

Note: Bit 0 (LSB) of Data byte 17 will always be 1b.

If the SMBus Address has been resolved (Address Resolved flag is true); for a general command the MNG will not acknowledge (NACK) this transaction, for a directed command the MNG will always acknowledge (ACK) this transaction.

This command will not affect the status or validity of the dynamic SMBus Address nor of the Address Resolved flag.

The command returns the UDID bytes as defined in Section 3.2.1.1.8.

S	Slave Address	Wr	A	Command	A	S	•••
	1100 001	0	0	See below	0		

7	1	1	8	1	
Slave Address	Rd	A	Byte Count	A	•••
1100 001	1	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data-1	A	Data-2	A	Data-3	A	Data-4	A	•••
UDID byte 15 (MSB)	0	UDID byte 14	0	UDID byte 13	0	UDID byte 12	0	

8	1	8	1	8	1	8	1	
Data-5	A	Data-6	A	Data-7	A	Data-8	A	•••
UDID byte 11	0	UDID byte 10	0	UDID byte 9	0	UDID byte 8	0	

8	1	8	1	8	1	
Data-9	A	Data-10	A	Data-11	A	•••
UDID byte 7	0	UDID byte 6	0	UDID byte 5	0	

8	1	8	1	8	1	8	1	
Data-12	A	Data-13	A	Data-14	A	Data-15	A	•••
UDID byte 4	0	UDID byte 3	0	UDID byte 2	0	UDID byte 1	0	

8	1	8	1	8	1	1
Data-16	A	Data-17	A	PEC	~Ã	P
UDID byte 0 (LSB)	0	Device Slave Address	0	[Data dependent value]	1	

10.3.3 Manageability Host Interface

10.3.3.1 HOST CSR Interface (Function 1/0)

The software device driver of function 0/1 communicates with the manageability block through CSR access. The manageability is mapped to address space 0x8800 to 0x8FFF on the slave bus of each function.

Note: Writing to address 0x8800 from function 0 or from function 1 is targeted to the same address in the RAM.

10.3.3.2 Host Slave Command Interface to Manageability

This interface is used by the software device driver for several of the commands and for delivering various types of data in both directions (Manageability-to-Host and Host-to-Manageability).

The address space is separated into two areas:

- Direct access to the internal ARC data RAM: The internal data RAM is mapped to address space 0x8800 to 0x8EFF. Writing/reading to this address space goes directly to the RAM.
- Control registers are located at address 0x8F00.

10.3.3.3 Host Slave Command Interface Low Level Flow

This interface is used for the external host software to access the manageability subsystem. Host software writes a command block or read data structure directly from the data RAM. Host software controls these transactions through a slave access to the control register.

The following flow shows the process of initiating a command to the manageability block:

1. The software device driver reads the control register and checks that the *Enable* bit is set.
2. The software device driver writes the relevant command block into the RAM area.
3. The software device driver sets the *Command* bit in the control register. Setting this bit causes an interrupt to the ARC (can be masked).
4. The software device driver polls the control register for the command bit to be cleared by hardware.
5. When manageability finishes with the command, it clears the command bit (if the manageability should reply with data, it should clear the bit only after the data is in the RAM area where the software device driver can read it).

If the software device driver reads the control register and the *SV* bit is set, then there is a valid status of the last command in the RAM. If the *SV* bit is not set, then the command has failed with no status in the RAM.

10.3.3.4 Host Slave Command Registers

10.3.3.4.1 Host Interface Control Register (CSR Address 0x8F00; AUX 0x0700)

This register operates along with the host software/firmware interface.

10.3.3.4.2 Firmware Status 0 (FWS0R) Register (CSR Address 0x8F0C; AUX 0x0702)

This register operates along with the host software/firmware interface.

10.3.3.4.3 Software Status Register (CSR Address 0x8F10; AUX 0x0703)

This register operates along with the host software/firmware interface.

10.3.3.5 Host Interface Command Structure

Table 10-27 describes the structure used by the host driver to send a command to manageability firmware using he host interface slave command interface.

Table 10-27. Host Driver Command Structure

#Byte	Description	Bit	Value	Description
0	Command	7:0	Command Dependent	Specifies which host command to process.
1	Buffer Length	7:0	Command Length	Command Data Buffer length: 0 to 252, not including 32 bits of header.
2	Default/Implicit Interface	0	Command Dependent	Used for commands might refer to one of two interfaces (LAN or SMBus). 0b = Use default interface. 1b = Use specific interface.
	Interface Number	1	Command Dependent	Used when bit 0 (Default/Implicit interface) is set: 0b = Apply command for interface 0. 1b = Apply command for interface 1. When bit 0 is set to 0b, it is ignored.
	Reserved	7:2	0x0	Reserved
3	Checksum	7:0	Defined Below	Checksum signature.
255:4	Data Buffer	7:0	Command Dependent	Command Specific Data Minimum buffer size: 0. Maximum buffer size: 252.

10.3.3.6 Host Interface Status Structure

Table 10-28 lists the structure used by manageability firmware to return a status to the host driver via the host interface slave command interface. A status is returned after a command has been executed.

Table 10-28. Status Structure Returned to Host Driver

#Byte	Description	Bit	Value	Description
0	Command	7:0	Command Dependent	Command ID.
1	Buffer Length	7:0	Status Dependent	Status buffer length: 252:0
2	Return Status	7:0	Depends on Command Executing Results	Defined in commands description.
3	Checksum	7:0	Defined Below	Checksum signature.
255:4	Data Buffer		Status Dependent	Status configuration parameters Minimum Buffer Size: 0. Maximal Buffer Size: 252.

10.3.3.7 Checksum Calculation Algorithm

The Host Command/Status structure is summed with this field cleared to 0b. The calculation is done using 8-bit unsigned math with no carry. The inverse of this sum is stored in this field (0b minus the result). Result: The current sum of this buffer (8-bit unsigned math) is 0b.

10.3.3.8 Host Slave Interface Commands

The host interface command that is supported is the fail-over configuration command (besides debug commands that will not be described in this document).

10.3.3.9 Fail-Over Configuration Host Command

This command is used to update the Fail-Over Configuration register.

Table 10-29. Command for Updating Fail-Over Configuration Register

Byte	Description	Bit	Value	Description
0	Command	7:0	0x26	Fail-over configuration command.
1	Buffer Length	7:0	0x4	Four bytes of the fail-over configuration register.
2		7:0	0x0	
3	Checksum	7:0		Checksum signature of the Host command.
7:4	Data Buffer	7:0	Fail-over configuration dwords	Fail-over register value. Byte 4 is byte 0 of the configuration register.

Following is the status returned on this command:

Table 10-30. Status Returned

Byte	Description	Bit	Value	Description
0	Command	7:0	0x26	Fail-over configuration command
1	Buffer Length	7:0	0x0	No data in return status
2	Return Status	7:0	0x1	0x1 for good status
3	Checksum	7:0		Checksum signature

10.3.3.10 Read Fail-Over Configuration Host Command

This command is used to read the Fail-Over Configuration register.

Table 10-31. Commands to Read the Fail-Over Configuration Register

Byte	Description	Bit	Value	Description
0	Command	7:0	0x27	Read Fail-Over Configuration command.
1	Buffer Length	7:0	0x0	No data attached to this command.
2		7:0	0x0	
3	Checksum	7:0		Checksum signature of the Host command.
7:4	Data Buffer	7:0	Fail-over configuration dwords	Fail-over register value. Byte 4 is byte 0 of the configuration register.

Following is the status returned on this command:

Table 10-32. States Returned

Byte	Description	Bit	Value	Description
0	Command	7:0	0x27	Fail-over configuration command
1	Buffer Length	7:0	0x4	No data in return status

Table 10-32. States Returned

2	Return Status	7:0	0x1	0x1 for good status
3	Checksum	7:0		Checksum signature
7:4	Data Buffer	7:0	Fail-over configuration dwords	Fail over register value. Byte 4 is byte 0 of the configuration register.

10.4 Manageability Receive Filtering

10.4.1 Overview and General Structure

This section describes the manageability receive packet filtering flow. The description applies to the each of the 82576 LAN ports. A packet that is received by the 82576 can have one of the following results:

- Discarded
- Sent to host memory
- Sent to the external BMC
- Sent to both the BMC and host memory

There are two modes of receive manageability filtering:

- Receive all – all receive packets are routed to the BMC in this mode. It is enabled by setting the RCV_TCO_EN bit (which enables packets to be routed to the BMC) and RCV_ALL bit (which routes all packets to the BMC) in the MANC register. This mode is used for debug only and should not be used for regular operation.
- Receive Filtering – In this mode only certain types of packets are directed to the manageability block. The BMC should set the RCV_TCO_EN bit together with the specific packet type bits in the manageability filtering registers. Note that the RCV_ALL bit must be cleared.

The default mode is that every packet that is directed to the BMC is not directed to host memory. The BMC can also configure the 82576 to direct certain manageability packets to host memory by setting the EN_MNG2HOST bit in the MANC register. It then needs to configure the 82576 to send manageability packets to the host (according to their type) by setting the corresponding bits in the MANC2H register.

The BMC controls the types of packets that it receives by programming the receive manageability filters. The software device driver can not write to the manageability filter registers. The table below lists the registers that are only written by the BMC.

Table 10-33. Registers Written by the BMC

Register	Functionality	When Reset
MANC	General configuration of the manageability filters	Internal_Power_On_Reset and Firmware Reset
MANC2H	Enables routing of manageability packets to host	Internal_Power_On_Reset and Firmware Reset
MFVAL	Indicates valid entries in the manageability filter registers	Internal_Power_On_Reset and Firmware Reset
MDEF[7:0] MDEF_EXT[7:0]	Configuration of manageability decision filters	Internal_Power_On_Reset and Firmware Reset
MMAH[3:0], MMAL[3:0]	Four Unicast MAC manageability addresses	Internal_Power_On_Reset
MAVTV[7:0]	Eight VLAN tag values	Internal_Power_On_Reset
MFUTP[7:0]	16 destination port values	Internal_Power_On_Reset

Table 10-33. Registers Written by the BMC

FTFT	Values and mask and length for four flex TCO filters	Internal_Power_On_Reset
MIPAF	IP address for manageability filtering	Internal_Power_On_Reset
METF	L2 EtherType values	Internal_Power_On_Reset

These registers are reset only on Internal_Power_On_Reset. Registers that enable filters or functionality are loaded from the EEPROM following a firmware reset. See Section 6.7 for description of their location in the EEPROM map.

The high-level structure of manageability filtering is done using three steps. The first 2 stages are shared with host filtering and are described in Section 7.1.2.

1. Packets are filtered by L2 criteria (MAC address and unicast/multicast/broadcast)
2. Packets are then filtered by VLAN if a VLAN tag is present
3. Packets are filtered by the manageability filters (port, IP, flex, etc.)

Some general rules:

- Fragmented packets are passed to manageability but not parsed beyond the IP header.
- Packets with L2 errors (CRC, alignment, etc.) are never forwarded to manageability.

Note: If the manageability unit uses a dedicated MAC address/VLAN tag, it should take care not to use L3/L4 decision filtering on top of it. Otherwise all the packets with the manageability MAC address/VLAN tag filtered out at L3/L4 will be forwarded to the host.

The manageability filtering stage combines some of the checks done at the previous stages with additional L3/L4 checks into a final decision whether to route a packet to the BMC. The following sections describe the manageability filtering done at layers L3 and L4, followed by the final filtering rules.

The filtering rules are created by programming the decision filters as described in Section 10.4.4.

10.4.2 L2 EtherType Filters

The manageability L2 EtherType filters allow filtering of receive packets based on the Layer 2 EtherType field. The L2 type field of incoming packets is compared against the EtherType filters programmed in the METF.EType (up to 4 filters) and the result is incorporated to the decision filters..

Each of the manageability EtherType filters can be configured as pass (“positive”) or reject (“negative”) polarity. When there is no match on its EtherType, and the polarity bit is set for a filter, that filter is enabled so as to participate in the decision process.

Note: In order for the reverse polarity mode to be effective and block certain type of packets, the ethertype filter should be part of all the enabled decision filters.

Examples for usages of the L2 EtherType filters are:

- Block routing of packets with the NC-SI EtherType from being routed to the management controller. The NC-SI EtherType is used for communications between the management controller on the NC-SI link and the 82576. Packets coming from the network are not expected to carry this EtherType and such packets are blocked to prevent attacks on the management controller.
- Determine the destination of 802.1X control packets. The 802.1X protocol is executed at different times in either the management controller or by the Host. The L2 EtherType filters are used to route these packets to the proper agent.

10.4.3 L3 and L4 Filters

10.4.3.1 ARP Filtering

The 82576 supports filtering of both ARP request packets (initiated externally) and ARP responses (to requests initiated by the BMC or the 82576).

10.4.3.2 Neighbor Discovery Filtering

The 82576 supports filtering of neighbor Discovery packets. neighbor Discovery uses the IPV6 destination address filters defined in the MIPAF registers (all enabled IPv6 addresses are matched for neighbor Discovery).

10.4.3.3 Port 0x298 / 0x26F Filtering

The 82576 supports filtering by fixed destination ports numbers, port 0x26F and port 0x298.

10.4.3.4 Flex Port Filtering

The 82576 implements 16 flex destination port filters. The 82576 directs packets whose L4 destination port matches the value of the respective word in the MFUTP registers. The BMC must ensure that only valid entries are enabled in the decision filters below.

10.4.3.5 Flexible 128 Bytes Filter (TCO Filters)

10.4.3.5.1 Overview

The Flexible 128 filters are a set of filters designed to allow dynamic filtering of received packets. These filters are part of the Manageability Receive Filters. The filters do not make a decision on the packet's destination. They participate in the Decision mechanism for each received packet (Section 10.4.4).

Each filter allows a flexible testing of the first 128 bytes of the packet against a given value. The filter also allows testing of specific bytes by defining a byte-wise mask on the filter.

The 82576 provides four flex TCO filters. Each filter looks for a pattern match within the 1st 128 bytes of the packet. The BMC must ensure that only valid entries are enabled in the decision filters.

Note: The flex filters are temporarily disabled when read or written by the host. Any packet received during a read or write operation is dropped. Filter operation resumes once the read or write access completes.

10.4.3.5.2 Structure

Each filter is composed of the following fields:

1. Flexible Filter length: This field indicates the number of bytes in the packet header that should be inspected. This field also indicates the minimal length of packets in order to be inspected by the filter. Packet below that length will not be inspected by the filter. Valid values for this field are: 8*n, where n=1…8.
2. Data: This is a set of up to 128 bytes comprising the values that the header bytes of each packet are tested against.
3. Mask: This is a set of 128 bits corresponding to the 128 data bytes that indicate for each corresponding byte if is tested against its corresponding byte.

Overall, each filter tests the first 128 bytes (or less) of a packet, where not necessarily all bytes must be tested.

10.4.3.5.3 Programming

Programming each filter is done using the below 2 commands (NC-SI or SMBus) in a sequential manner:

1. Filter Mask and Length. This command configures the following fields.
   a. Mask: A set of 16 bytes containing the 128 bits of the mask. Bit 0 of the first byte corresponds to the first byte on the wire.
   b. Length: A 1-byte field indicating the length.
2. Filter Data.
   The filter data is divided into groups of bytes. as described below:

Group	Test Bytes
0x0	0-29
0x1	30-59
0x2	60-89
0x3	90-119
0x4	120-127

Each group of bytes needs to be configured using a separate command, where the group number is given as a parameter.
The command has the following parameters:

a. Group number. A 1-byte field indicating the current group addressed

b. Data bytes. Up to 30 bytes of test-bytes for the current group

10.4.3.6 IP Address Filtering

The 82576 supports filtering by IP address through IPv4 and IPv6 address filters, dedicated to manageability. Two modes are possible, depending on the value of the MANC. EN_IPv4_FILTER bit:

- EN_IPv4_FILTER = 0b: the 82576 provides four IPv6 address filters.
- EN_IPv4_FILTER = 1b: the 82576 provides three IPv6 address filters and four IPv4 address filters.
- The MFVAL register indicates which of the IP address filters are valid (contains a valid entry and should be used for comparison).

10.4.3.7 Checksum Filter

If bit MANC.EN_XSUM_FILTER is set, the 82576 directs packets to the BMC only if they match all other filters previously described as well as pass L3/L4 checksum (if it exists).

10.4.4 Manageability Decision Filters

The manageability decision filters are a set of eight filters with the same structure. The filtering rule for each decision filter is programmed by the BMC and defines which of the L2, VLAN, and manageability filters participate in the decision. A packet that passes at least one rule is directed to manageability and possibly to the host.

The inputs to each decision filter are:

- Packet passed a valid management L2 unicast address filter.
- Packet is a broadcast packet.
- Packet has a VLAN header and it passed a valid manageability VLAN filter.
- Packet matched one of the valid IPv4 or IPv6 manageability address filters.
- Packet is a multicast packet.
- Packet passed ARP filtering (request or response).
- Packet passed neighbor Discovery filtering.
- Packet passed 0x298/0x26F port filter.
- Packet passed a valid flex port filter.
- Packet passed a valid flex TCO filter.
- Packet passed or failed an L2 EtherType filter.

The structure of each of the decision filters is shown in Figure 10-3 . A boxed “x.y” number indicates that the input is conditioned on a mask bit “y” defined in the MDEF and MDEF_EXT registers for this rule (where x=0 denotes MDEF and x=1 denotes MDEF_EXT). The decision filter rules are as follows:

- For a filter to be part of the decision process, at least one bit must be set in the respective MDEF and MDEF_EXT registers. If all bits are cleared (i.e. MDEF/MDEF_EXT=0x0000), then the decision filter is disabled and ignored.
- All enabled AND filters must match for the decision filter to match. An AND filter not enabled in the MDEF/MDEF_EXT registers is ignored.
- If at least one OR filter is enabled in the respective MDEF and MDEF_EXT registers, then at least one of the enabled OR filters must match for the decision filter to match. Otherwise, the OR filters are ignored in the decision (the filter might still match).

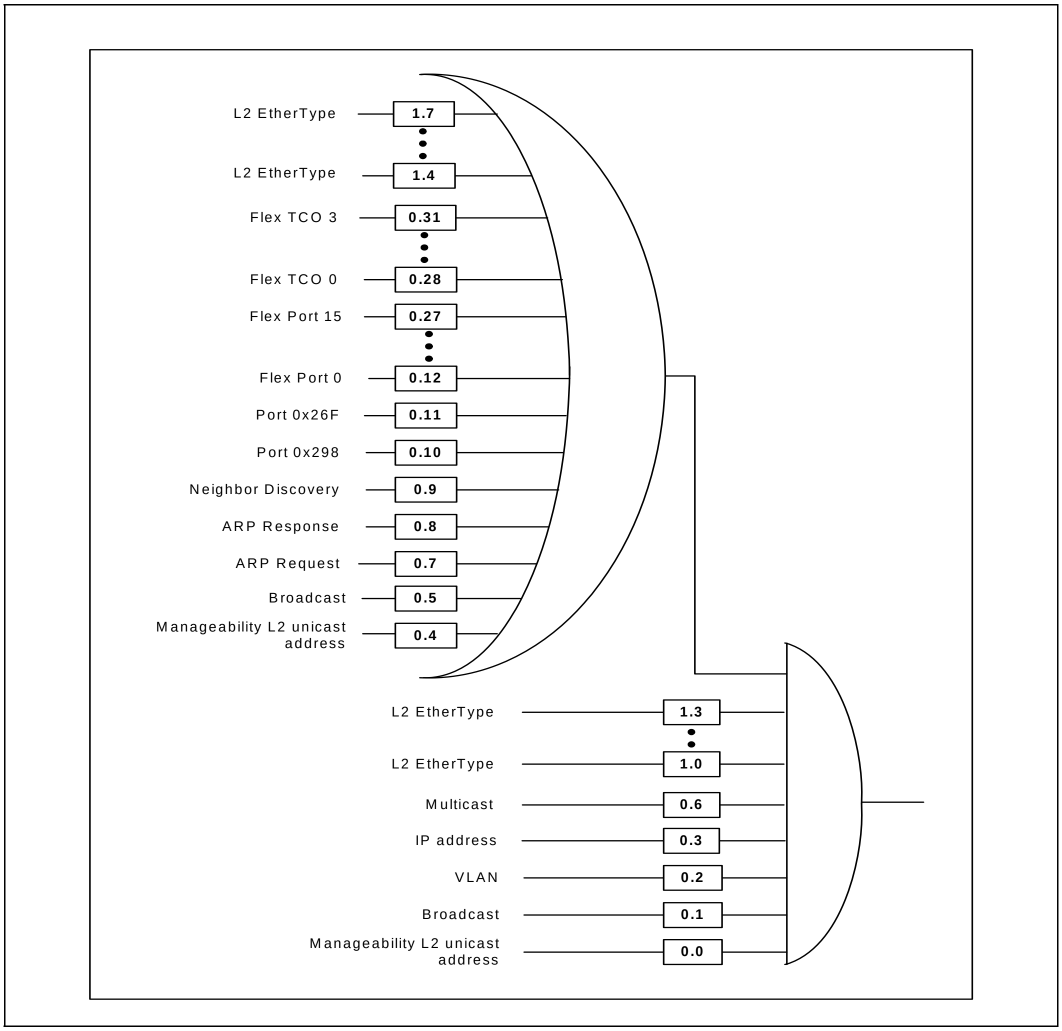


Figure 10-3. Manageability Decision Filters

A decision filter (for any of the 8 filters) defines which of the above inputs is enabled as part of the rule. The BMC programs two 32-bit registers per rule (MDEF[7:0] and MDEF_EXT[7:0]) with the settings as described in Table 10-34 and Table 10-35. A set bit allows its corresponding filter to participate in the filtering decision.

Table 10-34. Assignment of Decision Filters Bits

Filter	And / OR Input	Mask Bits in MDEF[7:0]
L2 Unicast Address	AND	0
Broadcast	AND	1
Manageability VLAN	AND	2
IP Address	AND	3
L2 Unicast Address	OR	4
Broadcast	OR	5
Multicast	AND	6

Table 10-34. Assignment of Decision Filters Bits (Continued)

ARP Request	OR	7
ARP Response	OR	8
neighbor Discovery	OR	9
Port 0x298	OR	10
Port 0x26F	OR	11
Flex Port 15:0	OR	27:12
Flex TCO 3:0	OR	31:28

Table 10-35. Assignment of Decision Filters Bits (MDEF_EXT)

Filter	And / OR Input	Mask Bits in MDEF_EXT[7:0]
L2 EtherType	AND	3:0
Reserved	--	7:4
L2 EtherType	OR	11:8
Reserved	--	31:12

The default mode is that every packet that is directed to the BMC is not directed to host memory. The BMC can also configure the 82576 to direct certain manageability packets to host memory by setting the EN_MNG2HOST bit in the MANC register and then configuring the 82576 to send manageability packets to the host, according to their type, by setting the corresponding bits in the MANC2H register (one bit per each of the eight decision rules).

All manageability filters are controlled by the BMC only and not by the LAN driver.

10.4.5 Possible Configurations

This section describes possible ways of using the management filters. Actual usage may vary.

10.4.5.1 Dedicated MAC Packet Filtering

- Select one of the eight rules for broadcast filtering.
- Set bit 0 of the decision rule to enforce MAC address filtering.
- Set other bits to qualify which packets are allowed to pass through. For example:
  - — Set bit 2 to qualify with manageability VLAN.
  - — Set bit 3 to qualify with a match to an IP address.
  - — Set any L3/L4 bits (30:7) to qualify with any of a set of L3/L4 filters.

10.4.5.2 Broadcast Packet Filtering

- Select one of the eight rules for broadcast filtering.
- Set bit 1 of the decision rule to enforce broadcast filtering.
- Set other bits to qualify which broadcast packets are allowed to pass through. For example:
  - — Set bit 2 to qualify with manageability VLAN.
  - — Set bit 3 to qualify with a match to an IP address.
  - — Set any L3/L4 bits (30:7) to qualify with any of a set of L3/L4 filters.

10.4.5.3 VLAN Packet Filtering

- Select one of the eight rules for VLAN filtering.
- Set bit two of the decision rule to enforce VLAN filtering.
- Set other bits to qualify which VLAN packets are allowed to pass through. For example:
  - — Set any L3/L4 bits (30:7) to qualify with any of a set of L3/L4 filters.

IPv6 filtering is done via the following IPv6-specific filters:

- IP Unicast filtering - requires filtering for Link Local address and a Global address. Filtering setup might depend on whether or not the MAC address is shared with the host or dedicated to manageability:
  - — Dedicated MAC address (for example, dynamic address allocation with DHCP does not support multiple IP addresses for one MAC address). In this case, filtering can be done at L2 using two dedicated unicast MAC filters.
  - — Shared MAC address (for example, static address allocation sharing addresses with host). In this case, filtering needs to be done at L3, requiring two IPv6 address filters, one per address.
- A neighbor Discovery filter - The 82576 supports IPv6 neighbor Discovery protocol. Since the protocol relies on multicast packets, the 82576 supports filtering of these packets. IPv6 multicast addresses are translated into corresponding Ethernet multicast addresses in the form of 33-33-xx-xx-xx-xx, where the last 32 bits of address are taken from the last 32 bits of the IPv6 multicast address. As a result, two direct MAC filters can be used to filter IPv6 solicited-node multicast packets as well as IPv6 all node multicast packets.

10.4.5.4 Receive Filtering with Shared IP - CPMP

When the BMC shares the MAC and IP address with the host, receive filtering is based mainly on identifying specific flows through port allocation. The following setting might be used:

- Select one of the eight rules.
- Set a manageability dedicated MAC filter to the host MAC address and set bit 0 in the MNG_ FILTER_RULE register.
- If VLAN is used for management, load one or more management VLAN filters and set bit 2 in the MNG_ FILTER_RULE register.
- ARP filter/neighbor Discovery filter is enabled when the BMC is responsible for handling the ARP protocol. Set bit 7 or bit 8 in the MNG_ FILTER_RULE register for this functionality.

§ §

11. Electrical / Mechanical Specification

11.1 Introduction

These specifications are subject to change without notice.

This chapter describes the 82576 DC and AC (timing) electrical characteristics. This includes absolute maximum rating, recommended operating conditions, power sequencing requirements, DC and AC timing specifications. The DC and AC characteristics include generic digital 3.3V IO specification as well as other specifications supported by the 82576.

For thermal information, see Chapter 15., "Thermal Design Specifications."

11.2 Operating Conditions

Table 11-1. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units
T_{case}	Case Temperature Under Bias	0	85	°C
$T_{storage}$	Storage Temperature Range	–65	140	°C
Vi/Vo	3.3V Compatible I/Os Voltage Analog 1.0 I/O Voltage Analog 1.8 I/O Voltage	Vss – 0.5 Vss – 0.2 Vss – 0.3	4.6 1.68 2.52	V
VCC3P3	3.3V Periphery DC Supply Voltage	Vss – 0.5	4.6	V
VCC	1.0V Core DC Supply Voltage	Vss – 0.2	1.68V	V
VCC1P8	1.8V Analog DC Supply Voltage	Vss – 0.3	2.52	V
VCC1P0	1.0V Analog DC Supply Voltage	Vss – 0.2	1.68V	V

Note: Ratings in this table are those beyond which permanent device damage is likely to occur. These values should not be used as the limits for normal device operation. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

11.2.1 Recommended Operating Conditions

Table 11-2. Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
Ta	Operating Temperature Range Commercial (Ambient; 0 CFS airflow)	0	55 70	°C	1,2,3

Notes:

1. For normal device operation, adhere to the limits in this table. Sustained operations of a device at conditions exceeding these values, even if they are within the absolute maximum rating limits, may

result in permanent device damage or impaired device reliability. Device functionality to stated DC and AC limits is not guaranteed if conditions exceed recommended operating conditions.

2. Recommended operation conditions require accuracy of power supply of +/-5% relative to the nominal voltage.
3. 70 value is for 1000BASE-T only.

11.3 Power Delivery

11.3.1 Power Supply specification

Table 11-3. External Power supply specification

VCC3P3 (3.3V) Parameters

Title	Description	Min	Max	Units
Rise Time	Time from 10% to 90% mark	0.1	100	mS
Monotonicity	Voltage dip allowed in ramp	N/A	0	mV
Slope	Ramp rate at any given time between 10% and 90% Min: 0.8*V(min)/Rise time (max) Max: 0.8*V(max)/Rise time (min)	24	28800	V/S
Operational Range	Voltage range for normal operating conditions	3	3.6	V
Ripple	Maximum voltage ripple (peak to peak)[1]	N/A	70	mV
Overshoot	Maximum overshoot allowed	N/A	100	mV
Overshoot Settling Time	Maximum overshoot allowed duration. (At that time delta voltage should be lower than 5mv from steady state voltage)	N/A	0.05	mS

[1] The ripple measurement should be performed at 20MHz BW

VCC1P8 (1.8V) Parameters

Title	Description	Min	Max	Units
Rise Time	Time from 10% to 90% mark	0.1	100	mS
Monotonicity	Voltage dip allowed in ramp	N/A	0	mV
Slope	Ramp rate at any given time between 10% and 90% Min: 0.8*V(min)/Rise time (max) Max: 0.8*V(max)/Rise time (min)	14	60000	V/S
Operational Range	Voltage range for normal operating conditions	1.71	1.89	V
Ripple	Maximum voltage ripple[1] (peak to peak)	N/A	40	mV
Overshoot	Maximum overshoot allowed	N/A	100	mV

Overshoot Settling Time	Maximum overshoot allowed duration. (At that time delta voltage should be lower than 5mv from steady state voltage)	N/A	0.1	mS
Decoupling Capacitance	Capacitance range	15	25	µF
Capacitance ESR	Equivalent series resistance of output capacitance	N/A	50	mΩ

1. The ripple measurement should be performed at 20MHz BW

VCC1P0 (1.0V) Parameters

Title	Description	Min	Max	Units
Rise Time	Time from 10% to 90% mark	0.1	100	mS
Monotonicity	Voltage dip allowed in ramp	N/A	0	mV
Slope	Ramp rate at any given time between 10% and 90% Min: 0.8*V(min)/Rise time (max) Max: 0.8*V(max)/Rise time (min)	7.6	33600	V/S
Operational Range	Voltage range for normal operating conditions	0.95	1.08	V
Ripple	Maximum voltage ripple (peak to peak)[1]	N/A	40	mV
Overshoot	Maximum overshoot allowed	N/A	100	mV
Overshoot Duration	Maximum overshoot allowed duration. (At that time delta voltage should be lower than 5mv from steady state voltage)	0.0	0.05	mS
Decoupling Capacitance	Capacitance range	15	25	µF
Capacitance ESR	Equivalent series resistance of output capacitance	5	50	mΩ

1. The ripple measurement should be performed at 20MHz BW

11.3.1.1 Power On/Off sequence

The following relationships between the rise time of the different power supplies should be maintained at all times when external power supplies are in use to avoid risk of either latch-up or forward-biased internal diodes:

T3.3 V supply ≤ T1.8 V supply ≤ T1.0 V supply

On power-on, after 3.3V reaches 10% of its final value, all voltage rails (1.8V and 1.0V) are allowed 100 ms to reach their final operating values. However, to keep leakage current at a minimum, it is recommended to turn on power supplies almost simultaneously (with delay between supplies at most a few milliseconds).

For power-down, it is recommended to turn off all rails at the same time and leave voltage to decay.

Table 11-4. Power Sequencing for the 82576

Symbol	Parameter	Min	Max	units
T_{3_18}	VCC3P3 (3.3V) stable to VCC1p8 stable	0	100	ms
T_{18_1}	VCC1p8 stable to VCC (1.0V) stable	0		ms

Table 11-4. Power Sequencing for the 82576

T_{3_1}	VCC3P3 (3.3V) stable to VCC (1.0V) stable	0	100	ms
Tm-per	3.3V core to PERST# de-assertion	100		ms
Tm-ppo	3.3V core to MAIN_PWR_OK on	0		ms

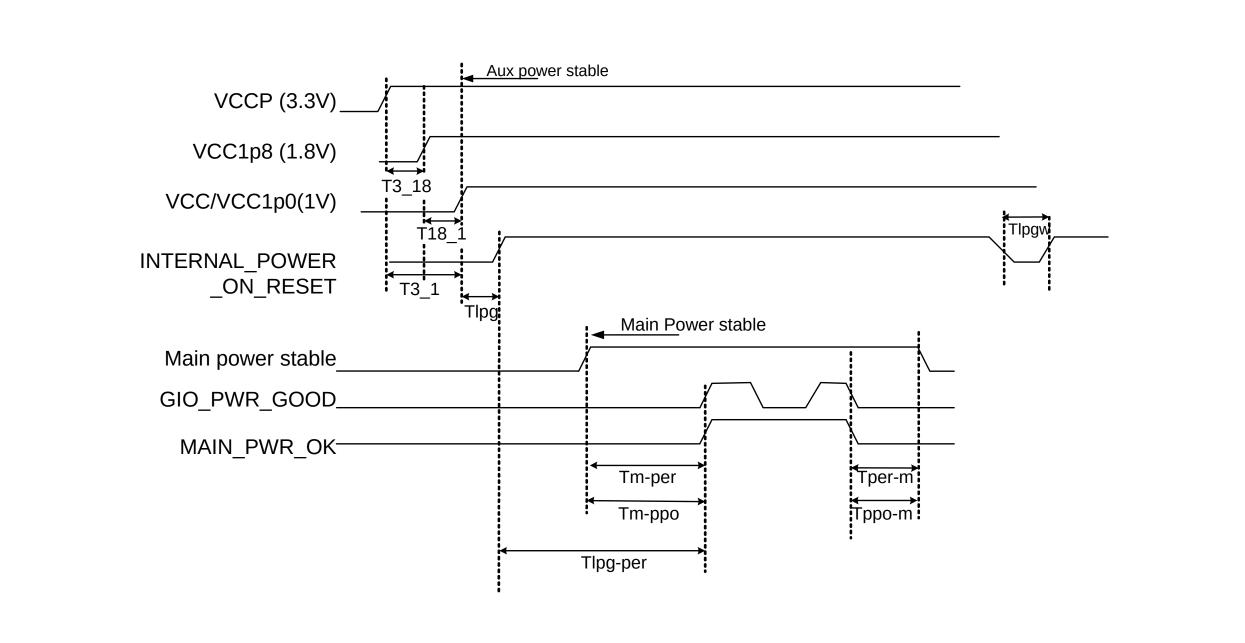


Figure 11-1. Power and Reset Sequencing

11.4 DC/AC specification

11.4.1 Ball Summary

See Chapter 2. for balls description and ball out map.

11.4.2 DC specifications

11.4.2.1 Current consumption

All the numbers in this section are based on A1 Class A measurements.

Table 11-5. Current Consumption Details

Condition	Speed (MBps)		3.3V (mA)	1.8V (mA)	1.0V (mA)	Total power (mW)
D0a - Active Link (manageability active)	10	Typ	15	330	300	943.5
	100	Typ	15	270	376	911.5
	1000 Copper	Typ	15	722	740	2089.1
		Max	15	744	1020	**2408.7**
	1000 SerDes	Typ	15	149	440	757.7
		Max	15	163	803	1145.9
D0a - Active Link (manageability off)	10	Typ	15	328	279	918.9
	100	Typ	15	270	355	890.5
	1000 Copper	Typ	15	722	711	2060.1
		Max	15	737	980	2356.1
	1000 SerDes	Typ	15	149	412	729.7
		Max	15	163	780	1122.9
D0a - Idle Link (manageability active) – L states Disabled	No link	Typ	15	108	352	595.9
	10	Typ	15	119	296	559.7
	100	Typ	15	270	371	906.5
	1000 Copper	Typ	15	722	692	2041.1
	1000 SerDes	Typ	15	149	412	729.7
D0a - Idle Link (manageability active) – L0s only	No link	Typ	15	87	351	557.1
	10	Typ	15	98	293	518.9
	100	Typ	15	248	369	864.9
	1000 Copper	Typ	15	701	692	2003.3
	1000 SerDes	Typ	15	128	408	687.9
D0a - Idle Link (manageability active) – L0s & L1	No link	Typ	15	64	288	452.7
	10	Typ	15	75	231	415.5
	100	Typ	15	224	308	760.7
	1000 Copper	Typ	15	677	623	1891.1
	1000 SerDes	Typ	15	105	349	587.5
D0a - Idle Link (manageability off) – L states Disabled	No link	Typ	15	108	265	508.9
	10	Typ	15	119	270	533.7
	100	Typ	15	268	340	871.9
	1000 Copper	Typ	15	722	663	2012.1
	1000 Fiber	Typ	15	149	378	695.7
D0a - Idle Link (manageability off) – L0s only	No link	Typ	15	108	265	508.9
	10	Typ	15	119	270	533.7
	100	Typ	15	268	340	871.9
	1000 Copper	Typ	15	700	657	1966.5
	1000 Fiber	Typ	15	128	375	654.9

Table 11-5. Current Consumption Details (Continued)

Condition	Speed (MBps)		3.3V (mA)	1.8V (mA)	1.0V (mA)	Total power (mW)
D0a - Idle Link (manageability off) – L0s & L1	No link	Typ	15	65	202	368.5
	10	Typ	15	75	206	390.5
	100	Typ	15	226	278	734.3
	1000 Copper	Typ	15	678	600	1869.9
	1000 Fiber	Typ	15	105	316	554.5
D3cold - wake-up enabled (manageability active)	No link	Typ	15	64	165	329.7
	10	Typ	15	75	175	359.5
	100	Typ	15	224	243	695.7
	1000 Fiber	Typ	15	104	279	515.7
D3cold - wake-up enabled (manageability off)	No link	Typ	15	64	87	251.7
	10	Typ	15	75	89	273.5
	100	Typ	15	224	153	605.7
	1000 Fiber	Typ	15	104	145	381.7
D3cold-wake disabled (manageability off)	No Link	Typ	15	68	88	259.9
D(r) Uninitialized Disabled through DEV_OFF_N	No Link	TYP	15	33	124	232.9

Notes: Typical conditions: room temperature (TA) = 25 C, nominal voltages and continuous network traffic at link speed at full duplex.

Maximum conditions: maximum operating temperature (TJ) values, typical voltage values and continuous network traffic at link speed at full duplex.

11.4.2.2 Digital I/O

Table 11-6. Digital IO DC Electrical Characteristics (Note 9)

Symbol	Parameter	Conditions	Min	Max	Units	Note
VCC3P3	Periphery supply		3.0	3.6	V	
VCC	Core supply		0.95	1.08	V	
VOH	Output High Voltage	IOH = -16mA; VCC3P3 = Min	2.4		V	
		IOH = -100µA; VCC3P3 = Min	VCC3P3–0.2			
VOL	Output Low Voltage	IOL = 13mA; VCC=Min		0.43	V	
		IOL = 100µA; VCC=Min		0.2	V	
		Iout = 13 mA		0.4	V	
VIH	Input High Voltage		2.0	VCC3P3 + 0.3	V	1
VIL	Input Low Voltage		-0.3	0.8	V	1
Iil	Input Current	VCC3P3 = Max; VI =3.6V/GND		24.5	µA	
IOFF	Current at IDDQ mode			50	µA	3
PU	Internal pullup		2.8	7.4	kΩ	4, 5, 6, 7
Ipup	Internal pull up current	0-0.5*VCC3P3[V]	0.43	1.1	mA	8

Table 11-6. Digital IO DC Electrical Characteristics (Note 9) (Continued)

	Built-in hysteresis		100	400	mV	
Vos	Overshoot		N/A	4	V	
Vus	Undershoot		N/A	-0.4	V	

Notes:

1. The input buffer also has hysteresis > 160mV
2. Pad Cin=5pF (max input capacitance), Cout=16pF (characterized max output load capacitance per 160MHz)
3. IDDQ mode maximum current consumption: CORE_VDD: 15µA; VCC3P3: 35µA
4. Internal pullup Max was characterized at slow corner (110C, VCC3P3=min, process slow); internal pullup Min was characterized at fast corner (0C, VCC3P3=max, process fast).
5. External R pull_down recommended is 400Ω
6. External R pull_up recommended is 3KΩ
7. External buffer recommended strength $\geq 2mA$
8. Internal pull-up max current consumption was characterized at fast corner(0C, VCC3P3=max, process fast)
    a. Internal pull-up min current consumption was characterized at slow corner(115C, VCC3P3=min, process slow)
9. Applies to PE_RSTn, LED0[3:0], LED1[3:0], SFP0_I2C_CLK, SFP1_I2C_CLK, SRDS0_SIG_DET, SRDS1_SIG_DET, DEV_OFF_N, M_PWR_OK, JTCK, JTDI, JTDO, JTMS, SDP0[3:0], SDP1[3:2], SDP1[0], FLSH_SI, FLSH_SO, FLSH_SCK, FLSH_CE_N, EE_DI, EE_DO, EE_SK, EE_CS_N.

11.4.2.3 Open Drain I/Os

Table 11-7. Open Drain DC Specifications (Note 1, 4)

Symbol	Parameter	Condition	Min	Max	Units	Note
VCC3P3	Periphery supply		3.0	3.6	V	
VCC	Core supply		0.9	1.32	V	
Vih	Input High Voltage		2.1		V	
Vil	Input Low Voltage			0.8	V	
Ileakage	Output Leakage Current	0 < Vin < VCC3P3		+/-17	µA	2
Vol	Output Low Voltage	@ Ipullup		0.4	V	4
Ipullup	Current sinking	Vol=0.4V	4		mA	
Cin	Input Pin Capacitance			7	pF	3
Cout	Output Pin Capacitance	F = 5 MHz		16	pF	3
Ioffsmb	Input leakage current	VCC3P3 off or floating		+/-10	µA	2

Note:

1. Applies to SMBD0, SMBCLK0, SMBALRT _N, PE_WAKE_N pads.
2. Device meets this whether powered or not.
3. Characterized, not tested.
4. OD no high output drive. VOL max=0.4V at 16mA, VOL max=0.2V at 0.1mA

11.4.2.4 NC-SI Input and Output Pads

Table 11-8. NC-SI Pads DC Specifications

Symbol	Parameter	Conditions	Min	Max	Units
VCC3P3	Periphery supply		3.0	3.6	V
VCC	Core supply		0.9	1.32	V
VOH	Output High Voltage	IOH = -4mA; VCC3P3 = Min	2.4		V
VOL	Output Low Voltage	IOL = 4mA; VCC3P3 = Min		0.4	V
VIH	Input High Voltage		2.0		V
VIL	Input Low Voltage			0.8	V
Vihyst	Input hysteresis		100		mV
Iil/Iih	Input Current	VCC3P3 = Max; Vin =3.6V/GND		15	µA
Ioll/Iohl	Output current	VCC = VOL/VOH. Low driving strength		4	mA
Iolh/Iohh	Output current	VCC = VOL/VOH. High driving strength		8	mA
Cin	Input Capacitance			5	pF
Ipup	Pull-Up current	Vout = 0V (GND)	0.4	1.3	mA
IOFF	Current at IDDQ mode	VCC3P3 (periphery)		80	µA

Note:

1. Applies to the NC-SI_ARB_OUT, NC-SI_CLK_OUT, NC-SI_CRS_DV, NC-SI_RXD[1:0], SDP1[1] - input/output pads and NC-SI_TX_EN, NC-SI_TXD[1:0], NC-SI_CLK_IN, NC-SI_ARB_IN - input pads.

11.4.3 Digital I/F AC Specifications

11.4.3.1 Digital I/O AC Specifications

Table 11-9. Digital I/O AC Electrical and Timing Characteristics

Parameters	Description	Min	Max	Cload	Note
Tor	Output Time rise	0.2ns	1ns	16pF	
Tof	Output Time fall	0.2ns	1ns		
Todr	Output delay rise	0.8ns	3ns		
Todf	Output delay fall	0.8ns	3ns		
Tidr	Input delay rise	0.3ns	1.5ns	200fF	1
Tidf	Input delay fall	0.3ns	1.5ns		1
Tir	Input Time rise	0.03ns	0.1ns		1
Tif	Input Time fall	0.03ns	0.1ns		1

Note: The input delay test conditions: Maximum input level = VIN = 2.7V; Input rise/fall time (0.2VIN to 0.8VIN) = 1ns (Slew Rate ~ 1.5ns).

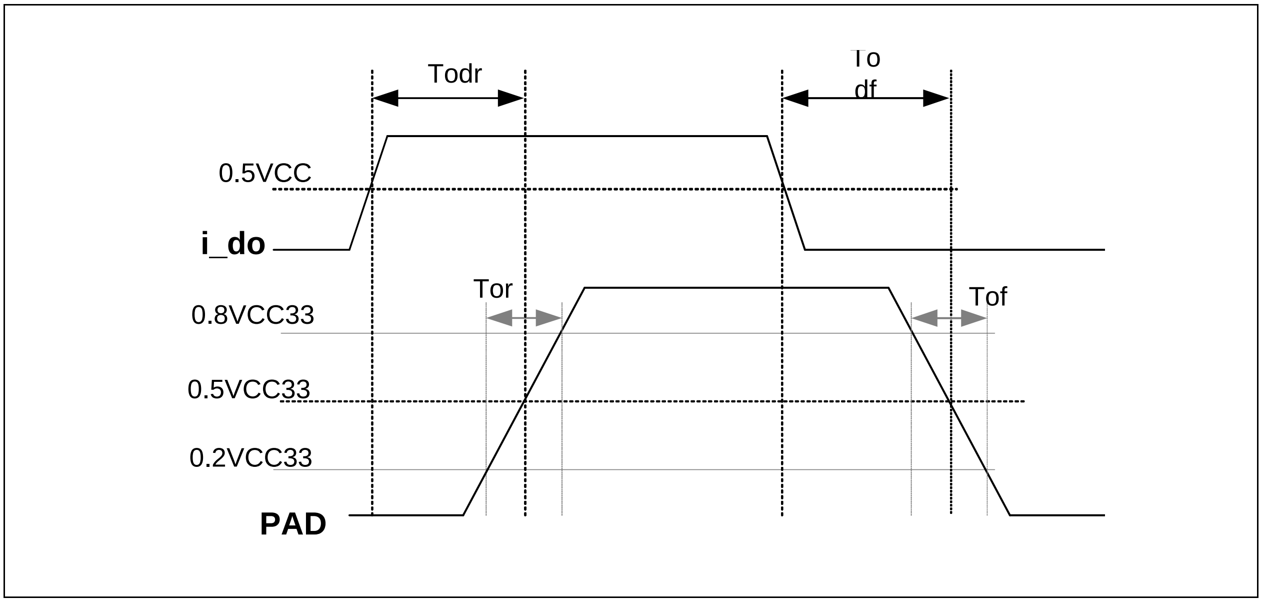


Figure 11-2. Digital I/O Output Timing Diagram

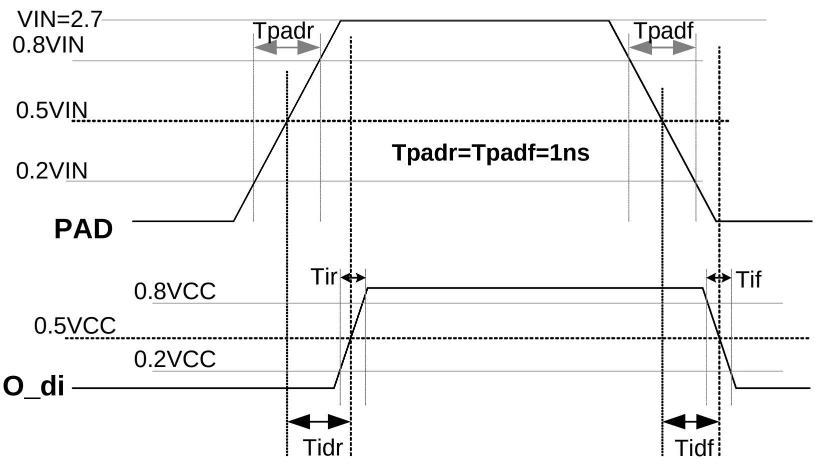


Figure 11-3. Digital IO Input Timing Diagram

11.4.3.2 Reset signals

The timing between the power up sequence and the different reset signals is described in Figure 4-2 and in Table 4-1.

11.4.3.2.1 Internal_Power_On_Reset

The 82576 uses an internal power on detection circuit in order to generate the Internal_Power_On_Reset signal. Reset can also be implemented when the external power on detection circuit determines that the device is powered up and asserts the Internal_Power_On_Reset signal to reset the device.

11.4.3.3 SMBus

The following table indicates the timing guaranteed when the driver or the agent is performing the action. Where only a typical value is specified, the actual value will be within 2% of the value indicated.

Table 11-10. SMBus Timing Parameters (Master Mode)

Symbol	Parameter	Min	Typ	Max	Units
F_{SMB}	SMBus Frequency		84	100	kHz
T_{BUF}	Time between STOP and START condition driven by the 82576		6.56		µs
$T_{HD:STA}$	Hold time after Start Condition. After this period, the first clock is generated.		6.72		µs
$T_{SU:STA}$	Start Condition setup time				µs
$T_{SU:STO}$	Stop Condition setup time		6.88		µs
$T_{HD:DAT}$	Data hold time		0.48		µs
$T_{TIMEOUT}$	Detect SMBClk low timeout	26.2		31.5	ms
T_{LOW}	SMBClk low time		5.76		µs
T_{HIGH}	SMBClk high time		6.56		µs

The following table indicates the timing requirements of the 82576 when it is the receiver of the indicated signal.

Many of these are below minimums specified by the SMBus specification.

Table 11-11. SMBus Timing Parameters (Slave Mode)

Symbol	Parameter	Min	Typ	Max	Units
F_{SMB}	SMBus Frequency	10		400	kHz
T_{BUF}	Time between STOP and START condition driven by the 82576.	1.44 *			µs
$T_{HD:STA}$	Hold time after Start Condition. After this period, the first clock is generated.	0.48 *			µs
$T_{SU:STA}$	Start Condition setup time	1.6 *			µs
$T_{SU:STO}$	Stop Condition setup time	1.76 *			µs
$T_{HD:DAT}$	Data hold time	0.32			µs
T_{LOW}	SMBClk low time	0.8 *			µs
T_{HIGH}	SMBClk high time	1.44 *			µs

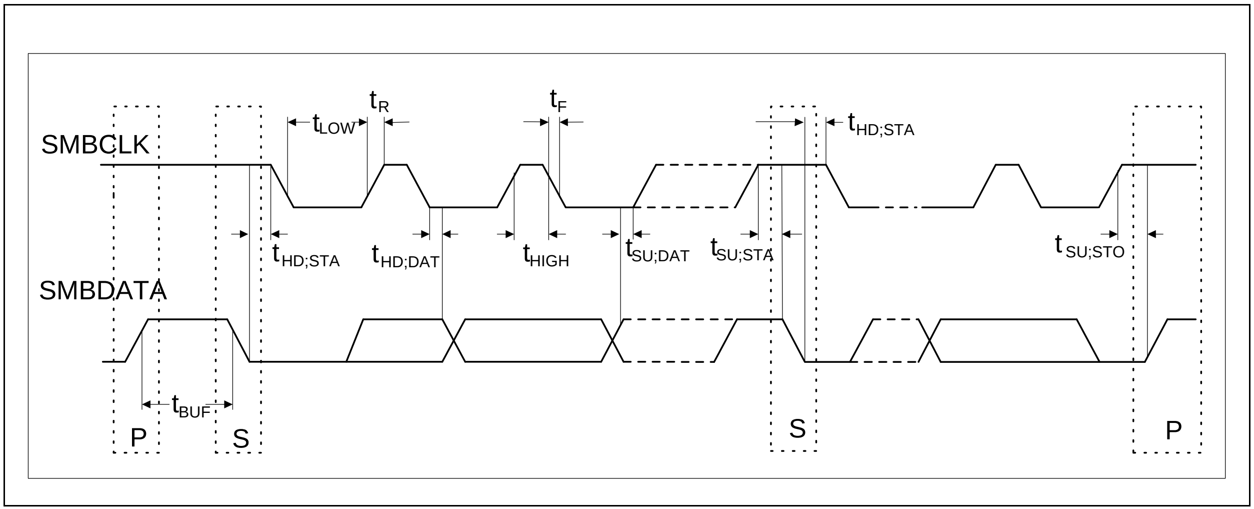


Figure 11-4. SMBus I/F Timing Diagram

11.4.3.4 FLASH AC specification

The 82576 is designed to support a serial flash. Applicable over the recommended operating range from Ta = -40C to +85C, VCC3P3 = 3.3V, Cload = 1 TTL Gate and 16 pF (unless otherwise noted). For FLASH I/F timing specification Table 11-12 and Figure 11-5.

Table 11-12. FLASH I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{SCK}	SCK clock frequency	0	15.625	20	MHz	[1]
t_{RI}	Input rise time		2.5	20	ns	
t_{FI}	Input fall time		2.5	20	ns	
t_{WH}	SCK high time	20	32		ns	[2]
t_{WL}	SCK low time	20	32		ns	[2]
t_{CS}	CS high time	25			ns	
t_{CSS}	CS setup time	25			ns	
t_{CSH}	CS hold time	25			ns	
t_{SU}	Data-in setup time	5			ns	
t_H	Data-in hold time	5			ns	
t_V	Output valid			20	ns	
t_{HO}	Output hold time	0			ns	
t_{DIS}	Output disable time			100	ns	

Note:

1. Clock is 62.5MHz divided by 450% duty cycle.

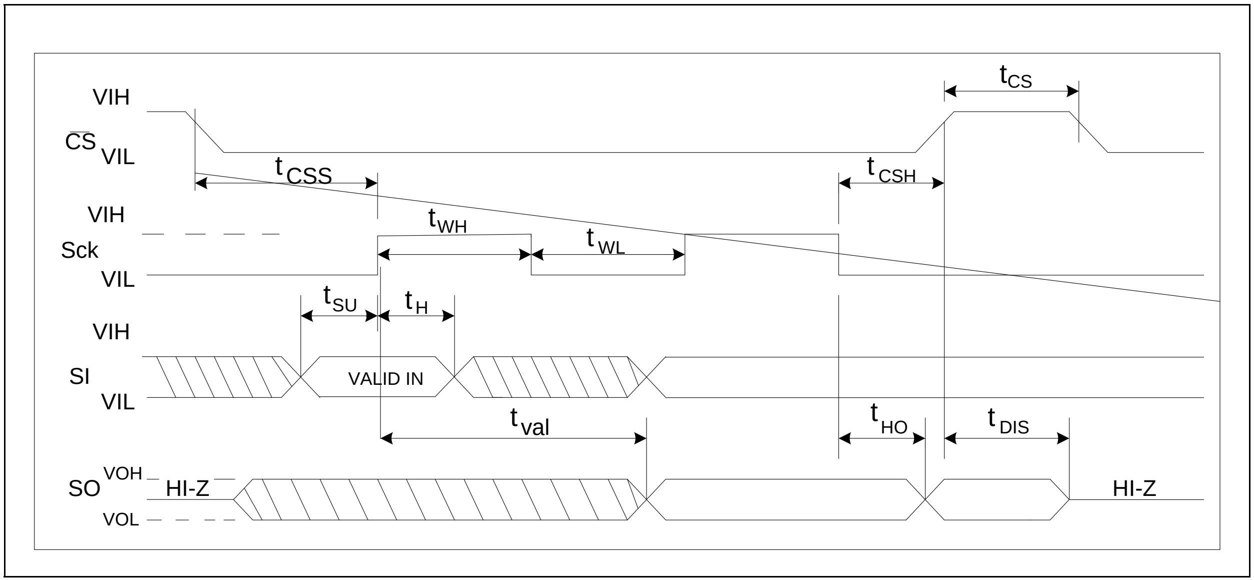


Figure 11-5. Flash Timing Diagram

11.4.3.5 EEPROM AC specification

The 82576 is designed to support a standard serial EEPROM. Applicable over recommended operating range from Ta = -40C to +85C, VCC3P3 = 3.3V, Cload = 1 TTL Gate and 16pF (unless otherwise noted). For EEPROM I/F timing specification see Table 11-13 and Figure 11-6.

Table 11-13. EEPROM I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{SCK}	SCK clock frequency	0	2	2.1	MHz	1
t_{RI}	Input rise time		2.5ns	2	µs	
t_{FI}	Input fall time		2.5ns	2	µs	
t_{WH}	SCK high time	200	250		ns	2
t_{WL}	SCK low time	200	250		ns	
t_{CS}	CS high time	250			ns	
t_{CSS}	CS setup time	250			ns	
t_{CSH}	CS hold time	250			ns	
t_{SU}	Data-in setup time	50			ns	
t_{H}	Data-in hold time	50			ns	
t_{V}	Output valid	0		200	ns	
t_{HO}	Output hold time	0			ns	
t_{DIS}	Output disable time			250	ns	

Notes:

1. Clock is 2MHz
2. 50% duty cycle

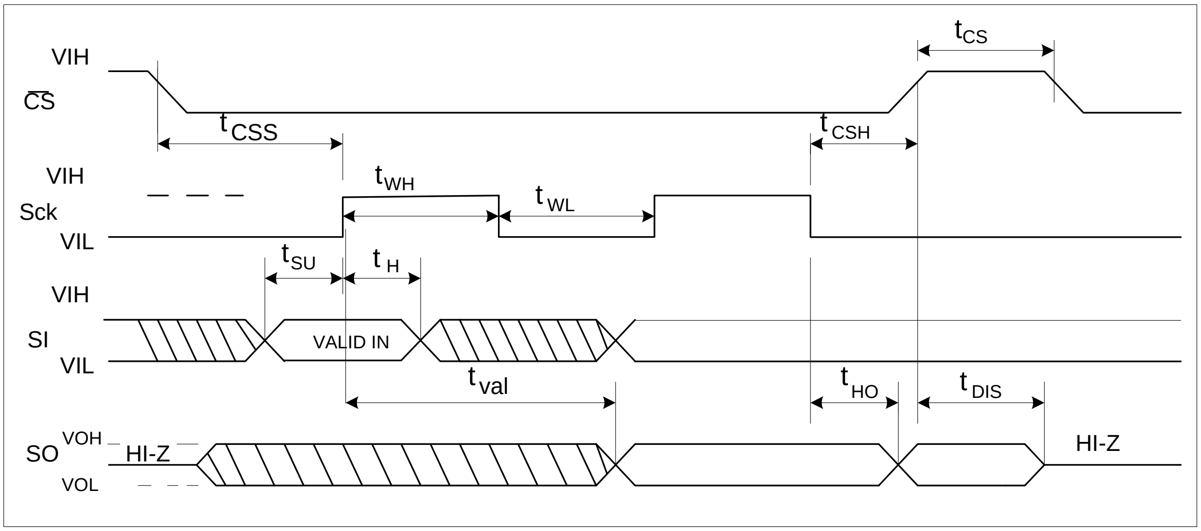


Figure 11-6. EEPROM Timing Diagram

11.4.3.6 NC-SI AC specification

The 82576 is designed to support the standard DMTF NC-SI interface. For NC-SI I/F timing specification see Table 11-14 and Figure 11-7.

Table 11-14. NC-SI AC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes
	REF_CLK Frequency		50		MHz	2
	REF_CLK Duty Cycle	35		65	%	1
	REF_CLK accuracy			100	ppm	
Tsu	TXD[1:0], TX_EN, Data Setup to REF_CLK rising edge	3			ns	
Thold	TXD[1:0], TX_EN Data hold from REF_CLK rising edge	1.5			ns	6
Tval	RXD[1:0], CRS_DV Data valid from REF_CLK rising edge	1.8		9	ns	
Tor	RXD[1:0], CRS_DV Output Time rise	0.5		6	ns	3
Tof	RXD[1:0], CRS_DV Output Time fall	0.5		6	ns	3
Tidr2	TXD[1:0], TX_EN Input delay rise	0.5		6	ns	4, 5
Tidf2	TXD[1:0], TX_EN Input delay fall	0.5		6	ns	4, 5
Tir	TXD[1:0], TX_EN Input Time rise	0.02		0.15	ns	4, 5
Tif	TXD[1:0], TX_EN Input Time fall	0.02		0.15	ns	4, 5

Notes:

1. Clock Duty cycle measurement: High interval measured from Vih to Vil points, Low from Vil to next Vih.
2. Clock interval measurement from Vih to Vih.
3. Cload = 25 pF.
4. Cload = 200 fF.

5. The input delay test conditions: Maximum input level = VIN = 2.7V; Input rise/fall time (0.2VIN to 0.8VIN) = 1ns (Slew Rate ~ 1.5ns).

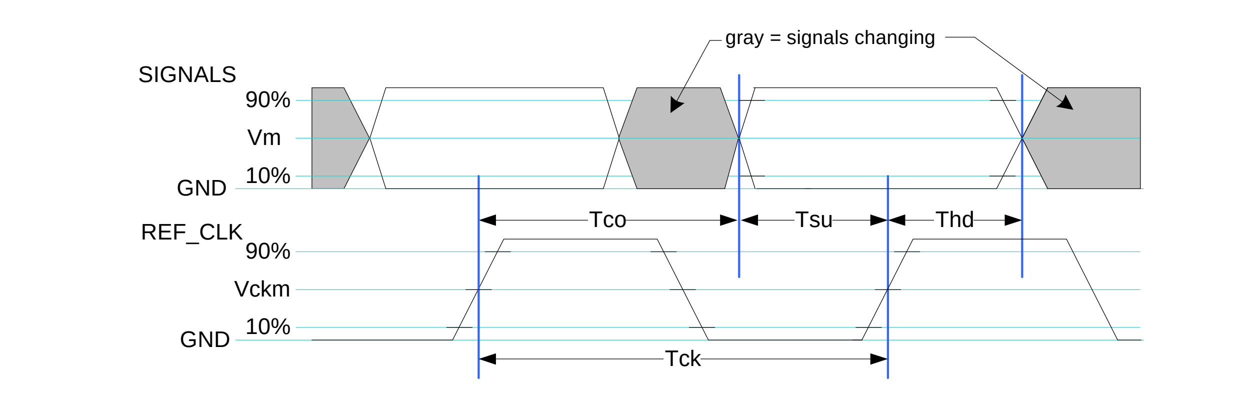


Figure 11-7. NC-SI Timing Diagram

11.4.3.7 JTAG AC specification

the 82576 is designed to support the IEEE 1149.1 standard. Following timing specifications are applicable over recommended operating range from Ta = 0^oC to $+70^oC$, VCC3P3 = 3.3V, Cload = 16pF (unless otherwise noted). For JTAG I/F timing specification see Table 11-15 and Figure 11-8.

Table 11-15. JTAG I/F timing parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{JCLK}	JTCK clock frequency			10	MHz	
t_{JH}	JTMS and JTDI hold time	10			nS	
t_{JSU}	JTMS and JTDI setup time	10			nS	
t_{JPR}	JTDO propagation Delay			15	nS	

Notes:

1. The table applies to JTCK, JTMS, JTDI and JTDO.
2. Timing measured relative to JTCK reference voltage of VCC3P3/2.

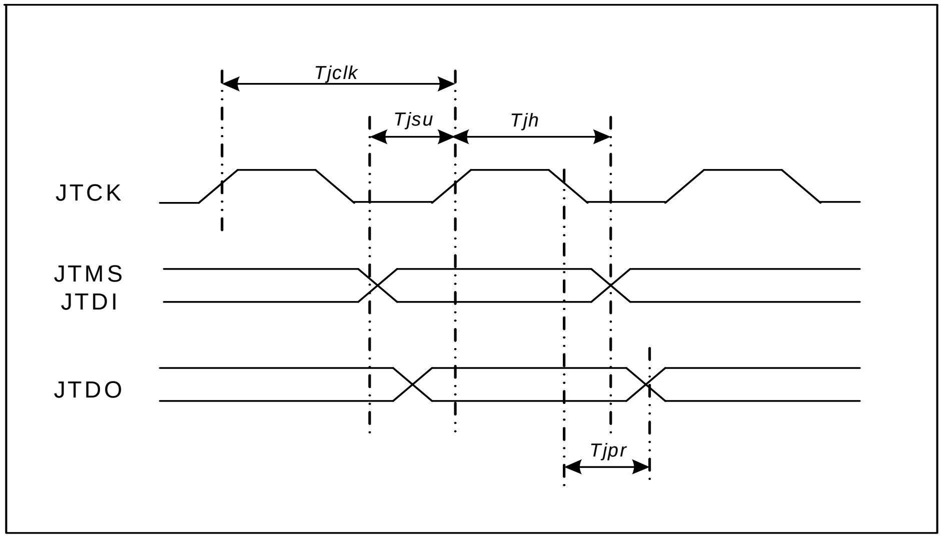


Figure 11-8. JTAG AC Timing Diagram

11.4.3.8 MDIO AC specification

the 82576 is designed to support the MDIO specifications defined in IEEE 802.3 clause 22. Following timing specifications are applicable over recommended operating range from Ta = 0^oC to $+70^oC$, VCC3P3 = 3.3V, Cload = 16pF (unless otherwise noted). For MDIO I/F timing specification see Table 11-16, Figure 11-9 and Figure 11-10.

Table 11-16. MDIO I/F timing parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{MCLK}	MDC clock frequency			2.5	MHz	
t_{MH}	MDIO hold time	10			nS	
t_{MSU}	MDIO setup time	10			nS	
t_{MPR}	MDIO propagation Delay	10		300	nS	

Notes:

1. The table above applies to MDIO0, MDC0, MDIO1, MDC1, MDIO2, MDC2, MDIO3, and MDC3.
2. Timing measured relative to MDC reference voltage of 2.0V (Vih).

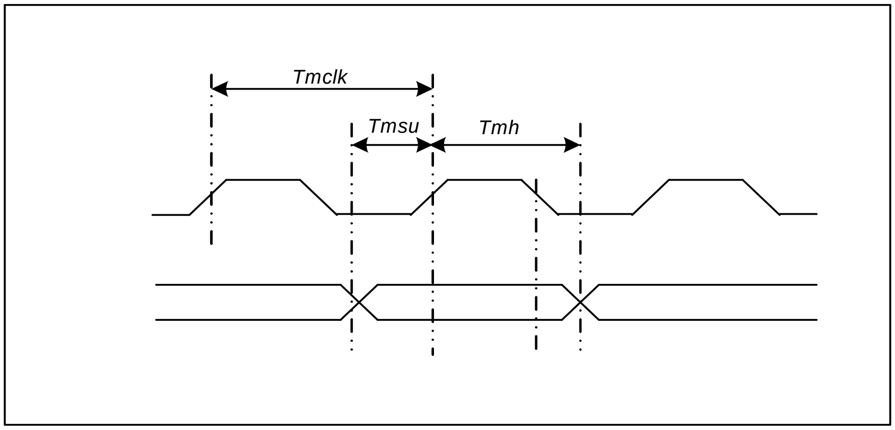


Figure 11-9. MDIO Input AC Timing Diagram

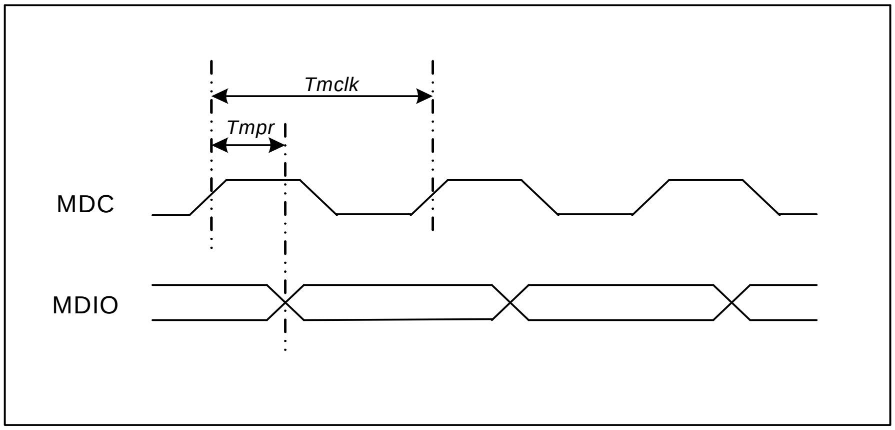


Figure 11-10. MDIO Output AC Timing Diagram

11.4.3.9 SFP 2 Wires I/F AC Specification

According to Atmel's AT24C01A/02/04 definition of the 2 wires I/F bus.

11.4.3.10 PCIe/SerDes DC/AC Specification

The transmitter and receiver specification are given per PCIe Card Electromechanical Specification rev 1.0.

11.4.3.11 PCIe Specification - Receiver

Specifications are from PCI Express* 2.0 (2.5Gbps) specification.

11.4.3.12 PCIe Specification - Transmitter

Specifications are from the PCI Express* 2.0 (2.5Gbps) specification.

11.4.3.13 PCIe Specification - Input Clock

The input clock for PCIe must be a differential input clock in frequency of 100 MHz. For full specifications please check the PCI-Express Card Electromechanical specifications (refclk specifications).

11.4.4 Serdes DC/AC Specification

The Serdes interface supports the PICMG 3.1 (1000BASE-BX), and SFP standards, this spec defines the interface for the back-plane board connection and the interface to fiber or SFP module.

11.4.4.1 Serdes Specification - Receiver

Specifications are from PICMG@ 3.1Draft specification Rev 1.0 1000Base-BX.

11.4.4.2 Serdes Specification - Transmitter

Specifications are from PICMG@ 3.1Draft specification Rev 1.0 1000Base-BX.

11.4.4.3 Serdes Specification -Input Clock

The input clock for Serdes is 25 MHz input crystal.

11.4.5 PHY Specification

DC/AC specification is according to Standard 802.3 and 802.3ab.

100 Base-T parameters are also described in standard ANSI X3.263.

11.4.6 XTAL/Clock specification

The 25 MHz reference clock of the 82576 can be supplied either from a crystal or from an external oscillator. The recommended solution is to use a crystal.

11.4.6.1 Crystal Specification

Table 11-17. Specification for External Crystal

Parameter Name	Symbol	Recommended Value	Conditions
Frequency	f_o	25.000 [MHz]	@25 [°C]
Vibration mode		Fundamental	
Cut		AT	
Operating /Calibration Mode		Parallel	
Frequency Tolerance @25°C	$\Delta f/f_o$ @25°C	±30 [ppm]	@25 [°C]
Temperature Tolerance	$\Delta f/f_o$	±30 [ppm]	
Operating Temperature	T_{opr}	-20 to +70 [°C]	
Non Operating Temperature Range	T_{opr}	-40 to +90 [°C]	

Table 11-17. Specification for External Crystal (Continued)

Equivalent Series Resistance (ESR)	R_s	50 [Ω] maximum	@25 [MHz]
Load Capacitance	C_{load}	20 [pF]	
Shunt Capacitance	C_o	6 [pF] maximum	
Pullability from Nominal Load Capacitance	$\Delta f/C_{load}$	15 [ppm/pF] maximum	
Max Drive Level	D_L	0.5 [mW]	
Insulation Resistance	IR	500 [MΩ] minimum	@ 100V DC
Aging	$\Delta f/f_o$	±5 [ppm/year]	
External Capacitors	C_1, C_2	27 [pF]	
Board Resistance	R_s	0.1 [Ω]	

11.4.6.2 External Clock Oscillator Specification

When using external oscillator the following connection must be used:

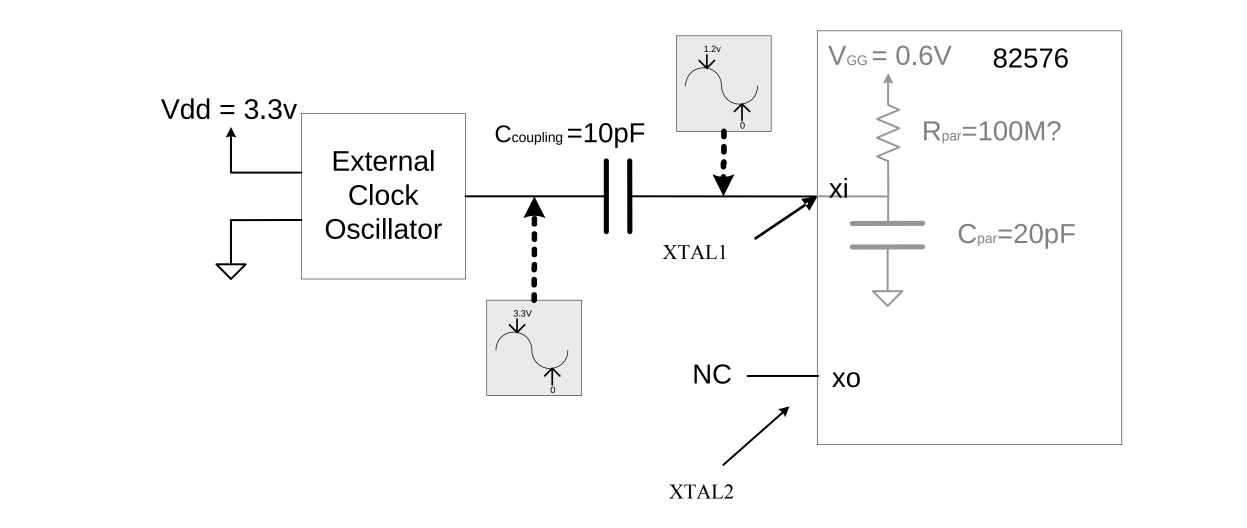


Figure 11-11. External Clock Oscillator Connectivity to the 82576.

Table 11-18. Specification for External Clock Oscillator

Parameter Name	Symbol	Value	Conditions
Frequency	f_o	25.0 [MHz]	@25 [°C]
External OSC Supply Swing	V_{p-p}	3.3 ± 0.3 [V]	
XTAL1 Swing	V_{SCP-P}	1.2 ± 0.1 [V]	
Frequency Tolerance	$\Delta f/f_o$	±50 [ppm]	-20 to +70 [°C]
Operating Temperature	T_{opr}	-20 to +70 [°C]	
Aging	$\Delta f/f_o$	±5 ppm per year	
Coupling capacitor	$C_{coupling}$	10 ± 2 [pF]	

11.4.7 RBIAS connection

For the PHY circuit, an external resistor of 1.4KΩ (accuracy 1%) is used as reference for the internal bias currents.

This resistor is connected to balls RBIAS0/1 and GND as described below.

Short connections for this resistor are compulsory.

Place the resistor as close as possible to the device (less than 1").

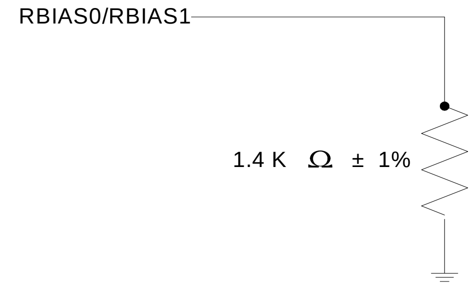


Figure 11-12. PHY RBIAS Connection

For the PCIe* & SerDes circuits, an external resistor of the same type is used as reference for the internal bias currents.

This resistor is connected to balls PE_RCOMP/SER_RCOMP and GND as described below.

Short connections for this resistor are compulsory.

Place the resistor as close as possible to the device (less than 1").

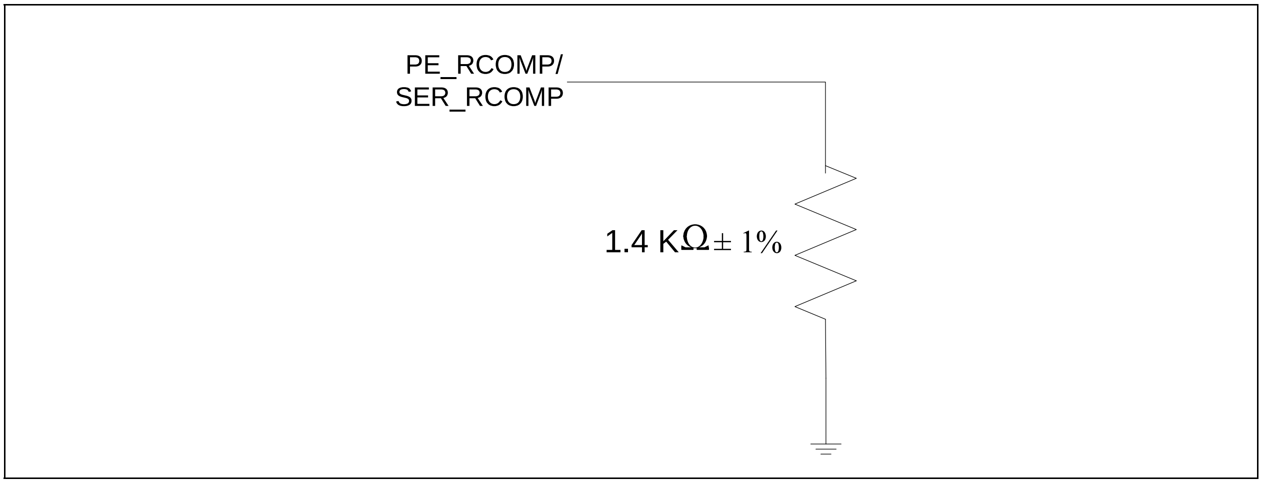


Figure 11-13. RCOMP Connection

11.5 EEPROM Flash Devices

While Intel does not make recommendations regarding these devices, the following devices have been used successfully in previous designs.

11.5.1 Flash

Type: SPI Flash

Size: 256 Kbytes (typical), depending on application

Table 11-19. Serial Flash Table

Density	Intel PN	Atmel PN	STM PN	SST PN
512KBit		AT25F512N-10SI-2.7	M25P05-AVMN6T	SST25VF512A
1MBit		AT25F1024N-10SI-2.7	M25P10-AVMN6T	SST25VF010A
2MBit		AT25F2048N-10SI-2.7	M25P20-AVMN6T	
4MBit		AT25F4096N-10SI-2.7	M25P40-AVMN6T	SST25VF040A
8MBit			M25P80-AVMN6T	SST25VF080A
16MBit	QB25F160S33T60 QB25F160S33B60 QH25F160S33T60 QH25F160S33B60 QB25F016S33T60 QB25F016S33B60 QH25F016S33T60 QH25F016S33B60		M25P16-AVMN6T	
32Mbit	QB25F320S33T60 QB25F320S33B60 QH25F320S33T60 QH25F320S33B60		M25P32-AVMN6T	

11.5.2 EEPROM Information

Table 11-20. EEPROM Devices

Density [Kbytes]	Atmel PN	STM PN	Catalyst PN
4	AT25040AN-10SI-2.7	M95040WMN6T	CAT25C040S-TE13
8	SI-2.7	M95080WMN6T	CAT25C080S-TE13
16	AT25160AN-10SI-2.7	M95160WMN6T	CAT25C16S-TE13
32	AT25320AN-10SI-2.7	M95320WMN6T	CAT25C32S-TE13
64	AT25640AN-10SI-2.7	M95640WMN6T	CAT25C64S-TE13
128	AT25128AN-10SI-2.7	M95128WMN6T	CAT25CS128-TE13
256	AT25256AN-10SI-2.7	M95256WMN6T	

11.6 Package Information

11.6.1 Mechanical

The 82576 is assembled into FCBGA5 package.

Table 11-21. Intel® 82576 GbE Controller Package Mechanical Specifications

Body Size	Ball Count	Ball Pitch	Ball Matrix
25mmx25mm	576	1.0 mm	24X24

11.6.2 Intel® 82576 GbE Controller Package Schematics

The following are package schematics.

THIS DRAWING CONTAINS INTEL CORPORATION CONFIDENTIAL INFORMATION. IT IS DISCLOSED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED, REPRODUCED, DISPLAYED OR MODIFIED, WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.

NOTES:

1. THIS DRAWING SHALL BE USED FOR ASSEMBLY REFERENCE AND FINISHED PROCESSOR MODULE FORM ONLY. CONSULT PIECE-PART/ SUBASSEMBLY DRAWINGS FOR APPLICABLE INFORMATION.
2. CRITICAL INTERFACE DIMENSIONS ARE NOTED WITH THIS SYMBOL ⟨C⟩.
3. DIMENSIONS/TOLERANCES FOR REFERENCE ONLY, REFER TO LOWER LEVEL DRAWINGS TO DETERMINE CONTROLLED SPECIFICATIONS.

△4 REFER TO INDIVIDUAL GENERIC MECHANICAL DRAWINGS FOR LAND PATTERN, PACKAGE THICKNESS, TOLERANCES, AND CHIP CAP KEEP-IN ZONES.

△5 REFER TO INDIVIDUAL GENERIC MECHANICAL DRAWINGS FOR HANDLING DEPOPULATION.

DWG. NO	SHT.	REV
A56807	1	19

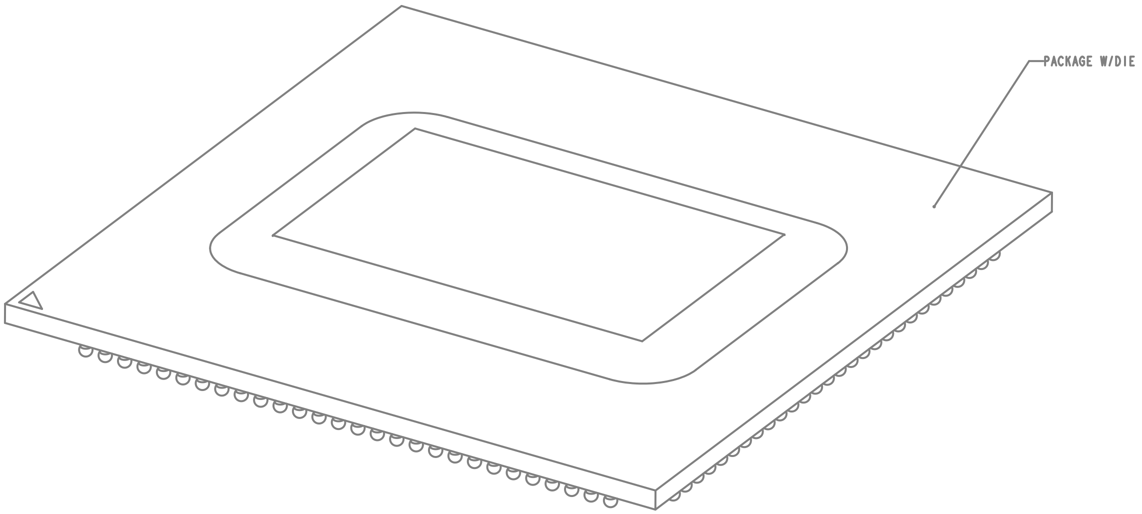
REVISION HISTORY

ZONE	REV	DESCRIPTION	DATE	APPROVED
-	1	PRELIMINARY RELEASE	04/05/01	SS/RP
S3-A4 S3-A4 S4-A5 S4-A6	2	ADD BALL PITCH COLUMN TO TABLE ADD 2 ROWS OF 1mm BALL PITCH INFO ADD (1.27mm BALL PITCH) TO 0.6±0.1 ADD 0.5±0.1 (1mm BALL PITCH)	07/30/01	SS/KTL/JC
-	3	REVISED PER DCN# 6003017 GES	08/22/02	JL/GS
-	4	REVISED PER DCN# 6065853 GES	02/06/03	JL/GS
-	5	REVISED PER ECO# 6245247 GES	12/08/03	JL/GS/JC
-	6	REVISED PER ECO# 6424719 JMC	11/17/04	BW/JC
-	7	REVISED PER ECO# 6535281 NW CHANGED DRAWING SIZE TO A1, WAS D	07/25/05	JL/NW
S3-C3 S3-C3 S3-B3 S4-A5	8	ADD (1.27mm BALL PITCH_297L) ONE ROW OF DIMENSION INFO ADD (1.27mm BALL PITCH_479L, 717L) D15504 ADD (1mm BALL PITCH_576L) "X" & "Y" DIMENSIONS ADD (E) ONE ROW OF DIMENSION INFO	10/20/05	AL/NW
-	9	REVISED PER ECO #6655047	03/09/06	BK/GX
S3-B1 S4-B4 -	10	ADD 1.016mm and 0.7mm REQUIREMENT TO TABLE ADD 1.016 and 0.7mm DETAIL TO ITEM B REVISED PER ECO #6743579	07/31/06	SL/AZLAN
S3-C3	11	ADDED 1.25mm BALL PITCH, 341L, PER ECO #6755045	08/24/06	AL/GS
S4 B4-B6	12	CHANGED ITEMS ON TABLE LINE FOR 0.7mm BALL PITCH: HEIGHT WAS 0.28 IS 0.35, DIA WAS 0.406 IS 0.457. PER ECO #6772182	09/20/06	SL/GS
-	13	REVISED PER ECO #6782182	10/10/06	GX/NW
S3	14	ADDED 0.8mm BALL PITCH, 88SL	11/20/06	AL/NW
S3-C3	15	ADDED 1MM BALL PITCH, 1155L ADDED 1.27MM BALL PITCH, 919L ADDED 1.092MM BALL PITCH, 1295L	01/16/07	FIZA
-	16	REVISED PER ECO #6833206	01/22/07	GX/NW
-	17	REVISED PER ECO #6908282	06/18/07	JL/MM
	18	REVISED PER ECO #7012952	02/29/08	FIZA
	19	REVISED PER ECO #7036731	04/29/08	BK/NS

UNLESS OTHERWISE SPECIFIED INTERPRET DIMENSIONS AND TOLERANCES IN ACCORDANCE WITH ASME Y14.5M-1994
DIMENSIONS ARE IN MILLIMETERS
ALL UNTOLERANCED LINEAR DIMENSIONS ±0.2
ANGLES ±1.0

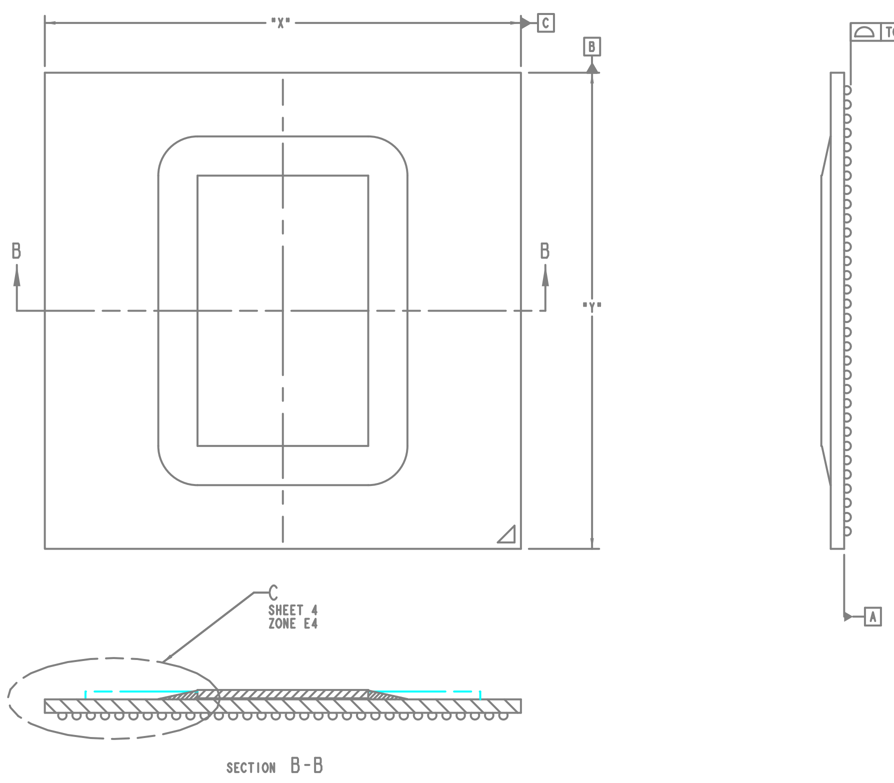
THIRD ANGLE PROJECTION

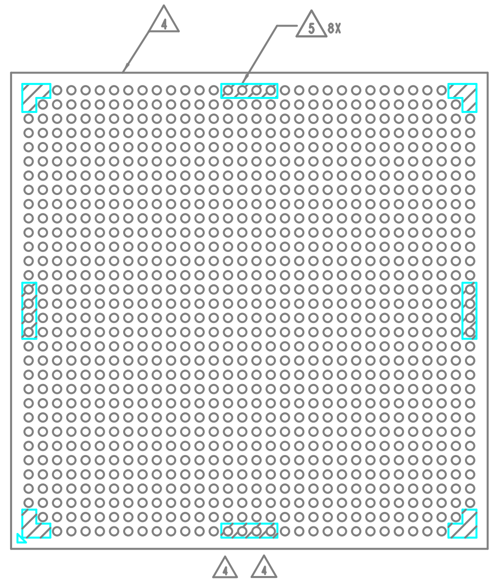
DESIGNED BY S SRINIVASAN	DATE 03/20/01	DEPARTMENT DPD/ATTD	intel 2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119
DRAWN BY R PULVINO	DATE 03/20/01	TITLE ASSEMBLY, FCBGAX, W/CUT IHS LID	
CHECKED BY	DATE		
APPROVED BY	DATE	SIZE A1	DRAWING NUMBER A56807 REV 19
MATERIAL SEE NOTES	FINISH SEE NOTES	SCALE: 4:1	DO NOT SCALE DRAWING SHEET 1 OF 4



DEPARTMENT		SIZE	DRAWING NUMBER	REV
DPD/ATTD	2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119	A1	A56807	19
		SCALE: 8:1	DO NOT SCALE DRAWING	SHEET 2 OF 4

THIS DRAWING CONTAINS INTEL CORPORATION CONFIDENTIAL INFORMATION. IT IS DISCLOSED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED, REPRODUCED, DISPLAYED OR MODIFIED, WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.

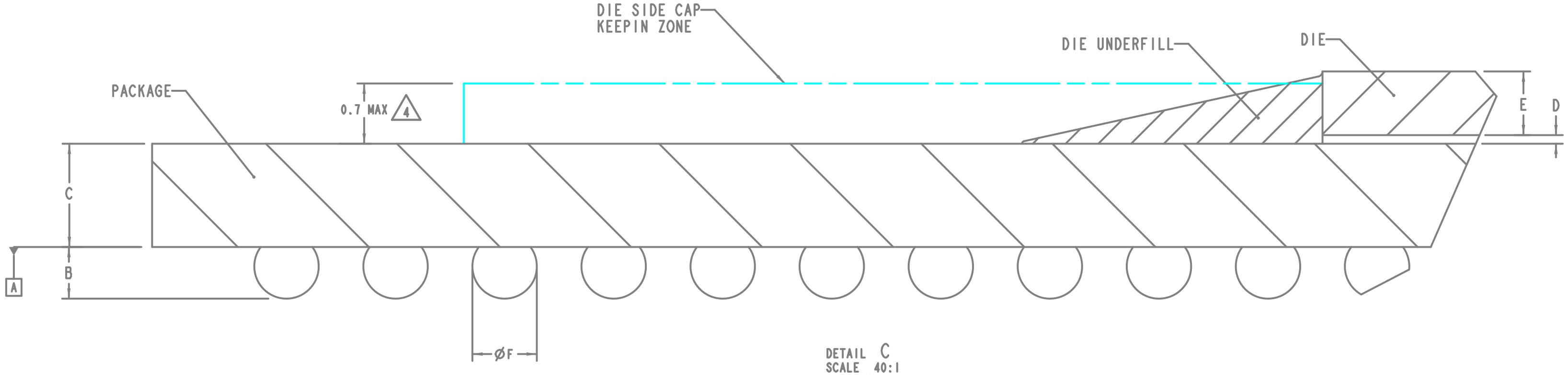




BALL PITCH (mm)	DESCRIPTION	"X"	"Y"	TOL "Z"	GEN. MECH. DRAWING #
1.016	1295L	37.5	37.5	0.203	D75163
1.27	297L	25	25		D36533
	567L	31	31		A35369, A95063, C11823
	479L, 717L	35	35		A14882, A97542, C26793, C61650, D15504, E14973
	593L, 733L, 829L	37.5	37.5		A06032
	919L	42.5	42.5		D76201
	1012L	42.5	42.5		A45509, A47401
	1005L, 1077L	42.5	42.5		A50293
1.25	341L	27	27		A70613
1.092	1295L	40	40		D75924
	1432L	42.5	42.5		A97508
1	196L	15	15		C19002
	256L	17	17		C23669
	484L	23	23		C23671
	576L	25	25		D17982
	883L	31	31		A95063
	1155L	35	35		C26793-003
	788L, 1356L, 1368L, 1357L, NON GRID	37.5	37.5		A67423, A74387
	NON GRID	40	37.5		C39714
	NON GRID	40	40		A99645
	971L, 1747L, 1763L, 1752L	42.5	42.5		A65307, A74758

BALL PITCH (mm)	DESCRIPTION	"X"	"Y"	TOL "Z"	GEN. MECH. DRAWING #
0.8	655L	24.5	19.5	0.203	C43803
	NON GRID	27	27		C65156
	885L	31	31		D76300
	NON GRID	34	34		C63452
	NON GRID	35	35		D24075, D31160
	NON GRID	37.5	37.5		C37619
0.7	NON GRID	27	27		E28508
	NON GRID	28	28		E23986
	NON GRID	34	34		E13023
	NON GRID	34	34		D59579
	NON GRID	35	35		D31160
0.6	NON GRID	28	28	0.1	E23986
	NON GRID	27	25		E24081
0.593	663L	14	19		D51112
	NON GRID	22	20		E25134
	1249L	22	22		D52303

DEPARTMENT			SIZE	DRAWING NUMBER	REV
DPD/ATTD	intel	2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119	A1	A56807	19
			SCALE: 5:1	DO NOT SCALE DRAWING	SHEET 3 OF 4



DETAIL C
SCALE 40:1
SHEET 3
ZONE B7

ITEM	DIMENSION (mm)	INCOMING BALL SIZE (mm)	DETAIL
B	0.7±0.1	0.889	1.27mm BALL PITCH
	0.6±0.1	0.762	
	0.5±0.1	0.6096	1mm, 1.016mm, 1.092mm and NON GRID BALL PITCH
	0.42±0.1	0.508	0.8mm BALL PITCH
	0.39±0.1	0.457	0.7mm BALL PITCH
	0.32±0.1	0.432	
	0.28±0.075	0.356	0.593mm BALL PITCH
C	△4	N/A	
D	0.1±0.025		Sn/Pb C4 BUMPS (x60, 859)
	0.085±0.025		Sn/Pb C4 BUMPS (861, 1263)
	0.065±0.025		Cu C4 BUMPS
E	0.74±0.025		200mm WAFER
	0.785±0.025		300mm WAFER WITHOUT SBSG
	0.754 ±0.028		300mm WAFER WITH SBSG
	0.815±0.025		1266 WAFER
F	0.52 ±0.01	0.457	0.7mm BALL PITCH

DEPARTMENT			SIZE	DRAWING NUMBER	REV
DPD/ATTD	intel	2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119	A1	A56807	19
			SCALE: 4:1	DO NOT SCALE DRAWING	SHEET 4 OF 4

DWG. NO	SHT.	REV
D17982	1	1

REVISION HISTORY

ZONE	REV	DESCRIPTION	DATE	APPROVED
	1	PRELLIMINARY RELEASE	04/05/05	QZ

NOTES:

1. ALL MATERIAL SHALL BE APPROVED BY INTEL.

2. DATA ENCLOSED IN PARENTHESES IS FOR REFERENCE ONLY.

△3 REFER TO DESIGN RULES FOR MAXIMUM DIE SIZE.

△4 HANDLING EXCLUSION ZONE. THERE SHALL BE NO COMPONENTS OR EPOXY ALLOWED. 1mm EDGE EXCLUSION ZONE APPLIES ON TONGUE SIDE OF EPOXY DISPENSE. FINAL TONGUE POSITION DEFINED IN PRODUCT ASSEMBLY DATA SHEET FOUND IN SPEED BOM OF SUBSTRATE PART NUMBER.

5. PACKAGE: FIBER REINFORCED RESIN CORE
WITH RESIN ONLY BUILD-UP LAYERS.

SOLDER RESIST: COLOR: GREEN
THICKNESS: SEE TABLE OF MATERIAL (SHEET 4)

COPPER THICKNESS: SEE TABLE OF MATERIALS (SHEET 4)

6. SURFACE FINFISH:

(ENIG) ELECTROLESS NICKEL GOLD
LAND AND FIDUCIAL NICKEL PLATING THICKNESS: 3um MIN.
LAND AND FIDUCIAL GOLD PLATING THICKNESS: 0.075um±0.045uM.

△7 SUCTION CUP HANDLING EXCLUSION ZONE (CROSS-HATCHED AREA). THERE SHALL BE NO COMPONENTS OR EXPOSED METAL ALLOWED EXCEPT FOR FIDUCIALS.

△8 INK SWATCH AREA: GREY INK THICNKESS: MIN 6.5um MAX 20um

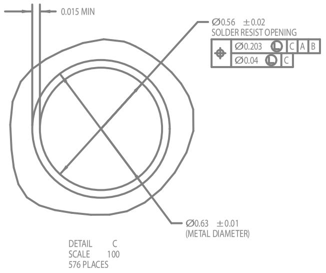
DASH NUMBER	INK SWATCH
-001	YES
-002	NO

-003	-002	-001	ITEM NO	PART NUMBER	DESCRIPTION
QTY PER ASSY					

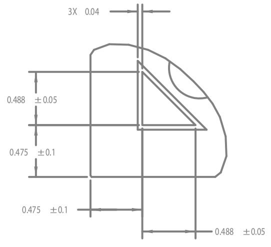
PARTS LIST

UNLESS OTHERWISE SPECIFIED INTERPRET DIMENSIONS AND TOLERANCES IN ACCORDANCE WITH ASME Y14.5M-1994 DIMENSIONS ARE IN MILLIMETERS TOLERANCES: .X ± 0.2 Angles ± 0.2 ° .XX ± 0.13 .XXX ± 0.05	DESIGNED BY Q. ZHOU	DATE 04/05/05	DEPARTMENT
THIRD ANGLE PROJECTION	DRAWN BY J. LAIRD	DATE 04/05/05	intel® CORP. 2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119
	CHECKED BY	DATE	TITLE GENERIC MECHANICAL, FCBGA5, 25mm X 25mm, 576 LANDS
	APPROVED BY	DATE	
	MATERIAL: SEE NOTE 5		SIZE D · CAGE CODE 34649 · DRAWING NUMBER D17982 · REV 1
	FINISH: SEE NOTE 6		SCALE: NONE · DO NOT SCALE DRAWING · SHEET 1 OF 4

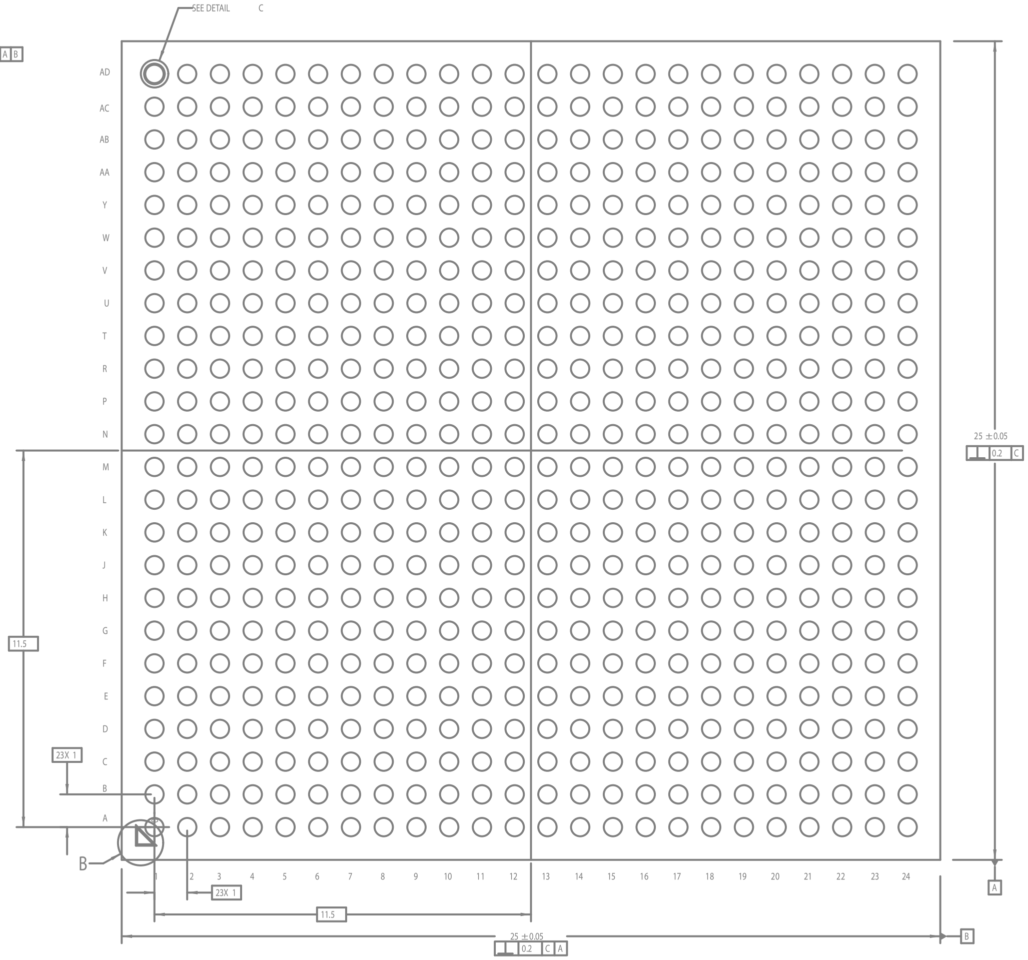
THIS DRAWING CONTAINS INTEL CORPORATION CONFIDENTIAL INFORMATION. IT IS DISCLOSED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED, REPRODUCED, DISPLAYED OR MODIFIED, WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.



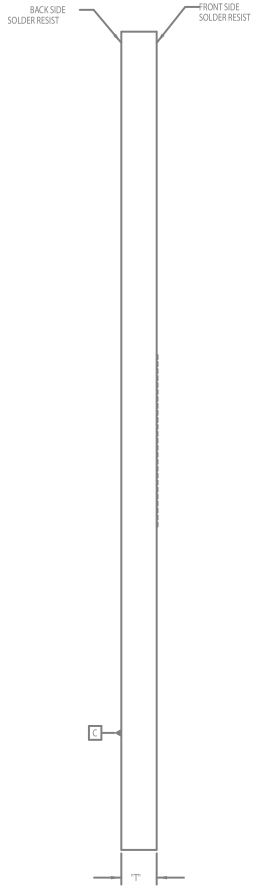
DETAIL C
SCALE 100
576 PLACES



DETAIL B
SCALE 50
PIN #1 ID



BOTTOM/LAND SIDE

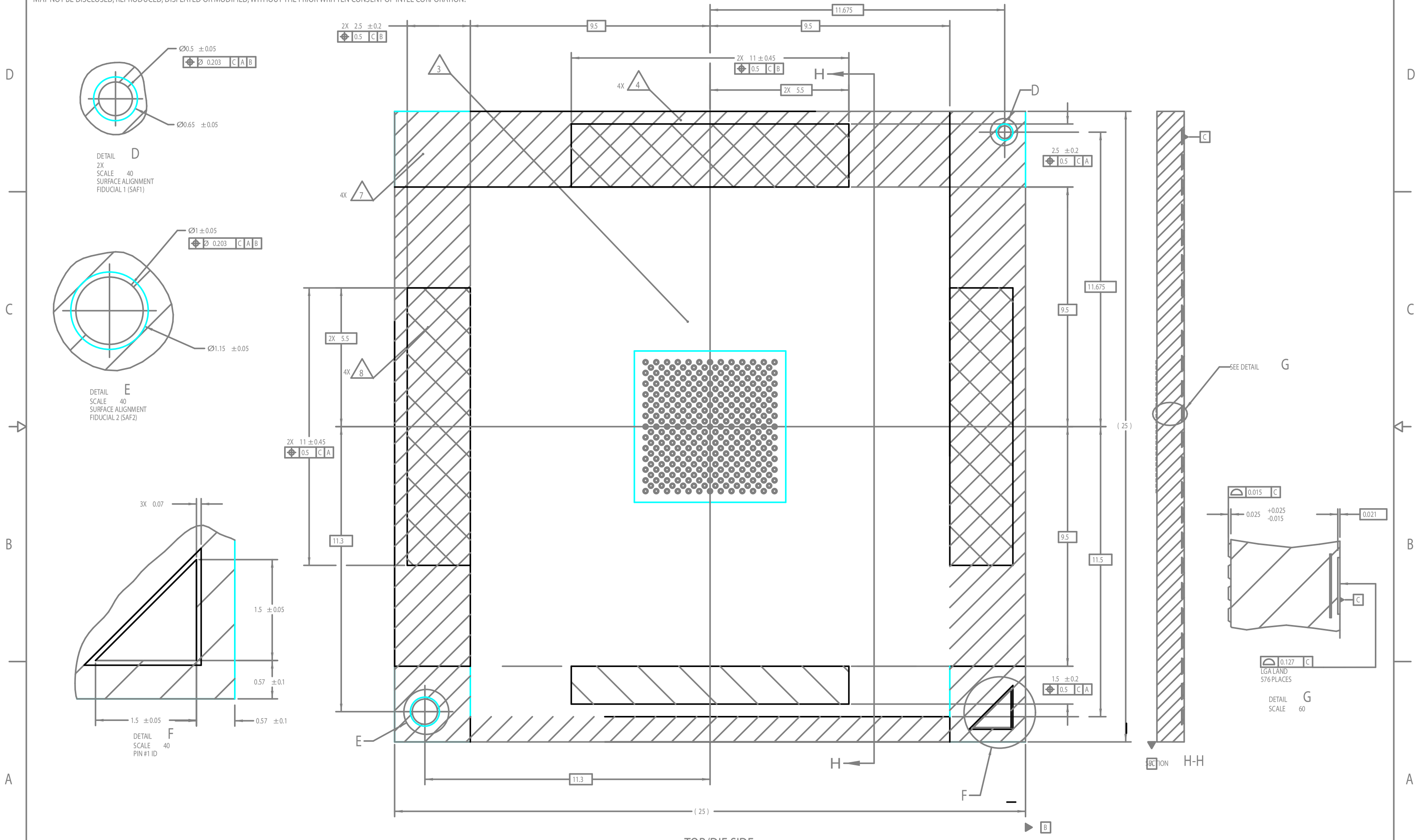


LAYER COUNT	DIMENSION "T"
4	0.99±0.06
6	1.08±0.06
8	1.17±0.085

DEPARTMENT
intel CORP.
2200 MISSION COLLEGE BLVD.
P.O. BOX 58119
SANTA CLARA, CA 95052-8119

SIZE	CAGE CODE	DRAWING NUMBER	REV
D	34649	D17982	1
SCALE: 15		DO NOT SCALE DRAWING	SHEET 2 OF 4

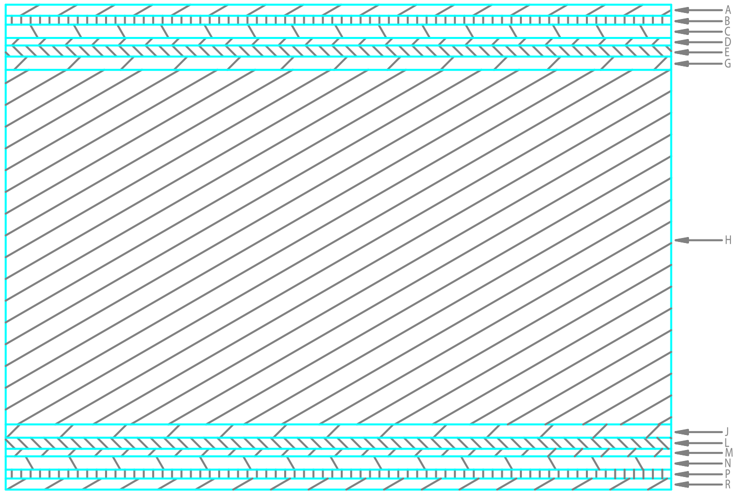
THIS DRAWING CONTAINS INTEL CORPORATION CONFIDENTIAL INFORMATION. IT IS DISCLOSED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED, REPRODUCED, DISPLAYED OR MODIFIED, WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.



TOP/DIE SIDE

DEPARTMENT	intel CORP. 2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119	SIZE D	CAGE CODE 34649	DRAWING NUMBER D17982	REV 1
		SCALE: 15		DO NOT SCALE DRAWING	SHEET 3 OF 4

THIS DRAWING CONTAINS INTEL CORPORATION CONFIDENTIAL INFORMATION. IT IS DISCLOSED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED, REPRODUCED, DISPLAYED OR MODIFIED, WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.



STACKUP DETAIL

TABLE OF MATERIALS				
FCBGA5 (4-8 LAYERS)				
CODE	LAYER #	DESCRIPTION	THICKNESS (mm)	TOLERANCE +/- (mm)
A		FRONT SOLDER RESIST	0.021	0.0075
	2F AND HIGHER	FOR EACH ADDITIONAL BUILDUP LAYER, REPEAT 1-2F AND 2F LAYERS		
D	2F	COPPER	0.015	0.005
E	1-2F	DIELECTRIC	0.03	0.006
G	1F	COPPER	0.025 (0.023 IF PTH'S ARE UNCAPPED)	0.005
H		INNER CORE DIELECTRIC	0.8	0.05
J	1B	COPPER	0.025 (0.023 IF PTH'S ARE UNCAPPED)	0.005
L	1-2B	DIELECTRIC	0.03	0.006
M	2B	COPPER	0.015	0.005
	2B AND HIGHER	FOR EACH ADDITIONAL BUILDUP LAYER, REPEAT 1-2B AND 2B LAYERS		
R		BACK SOLDER RESIST	0.021	0.0075

DEPARTMENT	intel® CORP. 2200 MISSION COLLEGE BLVD. P.O. BOX 58119 SANTA CLARA, CA 95052-8119	SIZE D	CAGE CODE 34649	DRAWING NUMBER D17982	REV 1
		SCALE: 4		DO NOT SCALE DRAWING	SHEET 4 OF 4

NOTE: This page intentionally left blank.

§ §

13. Diagnostics

13.1 Features

13.1.1 JTAG Test Mode Description

The 82576 includes a JTAG (TAP) port compliant with the *IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1 Specification*. The TAP controller is accessed serially through four dedicated pins: JTCK, JTMS, JTDI, and JTDO. JTMS, JTDI, and JTDO operate synchronously with JTCK. JTCK is independent of all other the 82576 clocks.

This interface can be used for test and debug purposes. System board interconnects can be DC tested using the boundary scan logic in pads. Table 13-1 shows TAP controller related pin descriptions. Table 13-2 describes the TAP instructions supported.

.

Table 13-1. TAP Controller Pins

Signal	I/O	Description
JTCK	In	Test clock input for the test logic defined by IEEE1149.1. If utilizing JTAG, connect to this signal ground through a 1 k ohm pull-down resistor.
JTDI	In	Test Data Input. Serial test instructions and data are received by the test logic at this pin. If utilizing JTAG, connect this signal to VCC33 through a 1 k ohm pull-up resistor.
JTDO	O/D	Test Data Output. The serial output for the test instructions and data from the test logic defined in IEEE1149.1. If utilizing JTAG, connect this signal to VCC33 through a 1 k ohm pull-up resistor.
JTMS	In	Test Mode Select input. The signal received at JTMS is decoded by the TAP controller to control test operations.

Table 13-2. TAP Instructions Supported

Instruction	Description	Comment
BYPASS	The BYPASS command selects the Bypass Register, a single bit register connected between TDI and TDO pins. This allows more rapid movement of test data to and from other components in the system.	IEEE 1149.1 Std. Instruction
EXTEST	The EXTEST Instruction allows circuitry or wiring external to the devices to be tested. Boundary-scan Register Cells at outputs are used to apply stimulus while Boundary-scan cells at input pins are used to capture data.	IEEE 1149.1 Std. Instruction
SAMPLE / PRELOAD	The SAMPLE/PRELOAD instruction is used to allow scanning of the boundary scan register without causing interference to the normal operation of the device. Two functions can be performed by use of the Sample/Preload instruction. SAMPLE – allows a snapshot of the data flowing into and out of a device to be taken without affecting the normal operation of the device. PRELOAD – allows an initial pattern to be placed into the boundary scan register cells. This allows initial known data to be present prior to the selection of another boundary-scan test operation.	IEEE 1149.1 Std. Instruction

Table 13-2. TAP Instructions Supported

IDCODE	The IDCODE instruction is forced into the parallel output latches of the instruction register during the Test-Logic-Reset TAP state. This allows the device identification register to be selected by manipulation of the broadcast TMS and TCK signals for testing purposes, as well as by a conventional instruction register scan operation. The ID code value for the 82576 A0 is 0x010C9013 (Intel's Vendor ID = 0x13, Device ID = 0x10C9, Rev ID = 0x0)	IEEE 1149.1 Std. Instruction
HIGHZ	The HIGHZ instruction is used to force all outputs of the device (except TDO) into a high impedance state. This instruction shall select the Bypass Register to be connected between TDI and TDO in the Shift-DR controller state.	IEEE 1149.1 Std. Instruction

§ §

14. Design Guidelines

This chapter provides guidelines for selecting components and connecting interfaces.

14.1 82575/82576

This section highlights implementation requirements necessary to support the 82575 using the same physical printed circuit board design. These devices have similar pin out requirements and functionality.

If a design does not implement a manageability (NC-SI) solution, then there are no special pin requirements for supporting both from a board perspective. The pin-out differences between the 82575 and the 82576 are only associated with the support for NC-SI multi-drop applications.

14.1.1 Pin Out Compatibility

The pin differences between the 82575 and 82576 affect the NC-SI interface used for the manageability features of the devices. The 82576 adds hardware arbitration support for systems that need NC-SI multi-drop. This results in a change of two pins.

Table 14-1 shows the two pins for the 82575 and the 82576 that have different functions. The effects of these differences are examined in the next section.

Table 14-1. 82575/82576 Pin Differences

82576 Pin Name	Ball Location	82575 Pin Name
NC-SI_ARB_IN	AD3	NCAD3
NC-SI_ARB_OUT	B3	NCB3

82576 Pin Descriptions:

- NC-SI_ARB_IN - NC-SI HW arbitration token input pin.
- NC-SI_ARB_OUT - NC-SI HW arbitration token output pin.

82575 Pin Descriptions:

- NCAD3 - No Connect pin requirement.
- NCB3 - No Connect pin requirement.

14.1.1.1 Printed Circuit Board Requirements

In order to design a board that supports both the 82575 and the Intel® 82576 GbE Controller, it is necessary to connect both AD3 and B3 balls to optionally stuff, 0-ohm resistors that could be connected to the platform for either 82575 or Intel® 82576 GbE Controller implementation.

If the NC-SI interface is not used in this particular application for both the 82575 and 82576, then leaving these pins as “no connects” is acceptable.

14.1.1.2 82576 Design

To enable a board design for the Intel® 82576 GbE Controller, stuff the 0-ohm resistors (connected to AD3 and B3) that could then be connected to the appropriate NC-SI arbitration pins of the platform. This is the only special requirement for implementing the 82576 silicon in a design to support both silicon devices.

14.1.1.3 82575 Design

To enable a board design for the 82575, DO NOT stuff the 0-ohm resistors (AD3 and B3) connections to the silicon. These pins are required as “No Connect” pins for the 82575.

14.2 Port Connection to the Device

This device implements signals required by the PCI Express* 2.0 (2.5Gbps) specification (PCIe*). This section provides an overview.

PCIe* is a dual simplex point-to-point serial differential low-voltage interconnect. The signaling bit rate is 2.5 Gbps per lane per direction. Each port consists of a group of transmitters and receivers located on the same chip. Each lane consists of a transmitter and a receiver pair. A link between the ports of two devices is a collection of lanes. The device supports up to four lanes on the PCIe interface.

Each signal is 8b/10b encoded with an embedded clock.

The topology consists of a transmitter (Tx) located on one device connected through a differential pair connected to the receiver (Rx) on a second device. The controller may be located on the motherboard or on an add-in card using a connector specified.

The lane is AC-coupled between its corresponding transmitter and receiver. The AC-coupling capacitor is located on the board close to transmitter side. Each end of the link is terminated on the die into nominal 100Ω differential DC impedance. Board termination is not required.

For more information, Section 3.1 in this manual and refer to the PCI Express Base Specification Revision 2.0 and PCI Express Card Electromechanical Specification Revision 2.0

14.2.1 PCIe Reference Clock

The device uses a 100 MHz differential reference clock, denoted PE_CLK_p and PE_CLK_n. This signal is typically generated on the system board and routed to the PCIe port. For add-in cards, the clock will be furnished at the PCIe connector.

The frequency tolerance for the PCIe reference clock is +/- 300 ppm.

14.2.2 Other PCIe Signals

The device implements other signals required by the specification. The Ethernet controller signals power management events to the system using the PE_WAKE_N signal, which operates very similarly to the familiar PCI PME# signal. There is a PE_RST_N signal which serves as the familiar reset function for the controller.

14.2.3 Physical Layer Features

14.2.3.1 Link Width Configuration

The device supports a maximum link width of x4, x2, or x1, refer to section Section 3.1.6.1.

14.2.3.2 Polarity Inversion

For details on PCI Express polarity inversion supported by the 82576 refer to section Section 3.1.6.2.

14.2.3.3 Lane Reversal

The following lane reversal modes are supported (see Figure 14-1):

- Lane configuration of x4, x2, and x1
- Lane reversal in x4 and in x2
- Degraded mode (downshift) from x4 to x2 to x1 and from x2 to x1, with one restriction - if lane reversal is executed in x4, then downshift is only to x1 and not to x2.

These restrictions require that a x2 interface to the 82576 GbE Controller must connect to lanes 0 &1 on the 82576 GbE Controller. The PCIe Card Electromechanical specification does not allow routing a x2 link to a wider connector. Therefore, the system designer is not allowed to connect a x2 link to lanes 2 and 3 of a PCIe connector. It is also recommended that, when using x2 mode on a network interface card, the 82576 GbE Controller be connected to lanes 0 & 1 of the card.

For further details on Lane reversal details refer to Section 3.1.6.5.

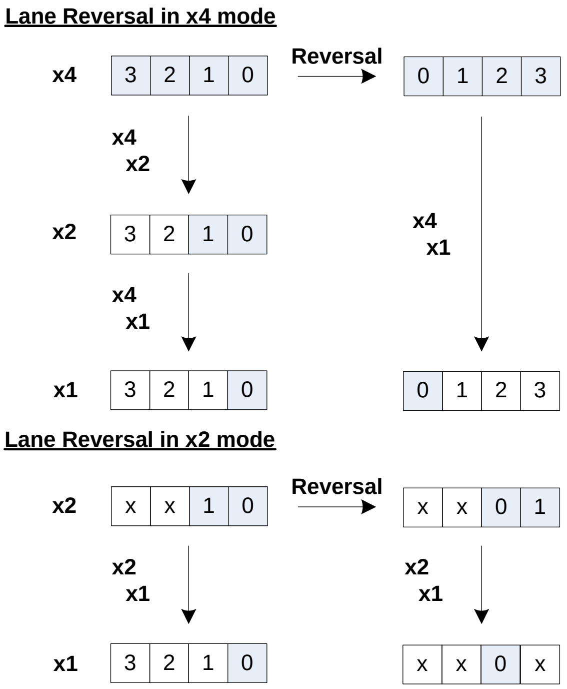


Figure 14-1. Lane Reversal Supported Modes

14.2.4 PCIe Routing

For information regarding the PCIe signal routing, please refer to the Intel PCIe Design Guide. Contact Intel for information.

14.3 Ethernet Component Design Guidelines

These sections provide recommendations for selecting components and connecting special pins.

For 1000 BASE-T designs, the main design elements are the 82576 Gigabit Ethernet Controller, an integrated discrete or magnetics module with RJ-45 connector, an EEPROM, and a clock source.

14.3.1 General Design Considerations for Ethernet Controllers

Follow good engineering practices with respect to unused inputs by terminating them with pull-up or pull-down resistors, unless the datasheet, design guide or reference schematic indicates otherwise. Do not attach pull-up or pull-down resistors to any balls identified as No Connect. These devices may access test modes that could be entered unintentionally.

14.3.1.1 Clock Source

All designs require a 25 MHz clock source. The 82576 Gigabit Ethernet Controller uses the 25 MHz source to generate clocks up to 125 MHz and 1.25 GHz for the PHY circuits, and 1.25 GHz for the SERDES. For optimum results with lowest cost, connect a 25 MHz parallel resonant crystal and appropriate load capacitors at the XTAL1 and XTAL2 leads. The frequency tolerance of the timing device should be 30 ppm or better. Refer to the application note, Intel Fast Ethernet Controllers Timing Device Selection Guide, AP-419, for more information on choosing crystals.

For information regarding the clock for the 82576, see the sections about frequency control (Section 14.3), crystals (Section 14.4), and oscillators (Section 14.5). When selecting a crystal as the clock source, refer to the crystal layout considerations in Section 14.7.1.2.1.

14.3.1.2 Magnetics for 1000 BASE-T

Magnetics for the 82576 can be either integrated or discrete.

The magnetics module has a critical effect on overall IEEE and emissions conformance. Occasionally, components that meet basic specifications may cause the system to fail IEEE testing because of interactions with other components or the printed circuit board itself. Carefully qualifying new magnetics modules prevents this problem.

When using discrete magnetics it is necessary to use Bob Smith termination: use four 75 Ω resistors for cable-side center taps and unused pins. This method terminates pair-to-pair common mode impedance of the CAT5 cable.

Use an EFT capacitor attached to the termination plane. Suggested values are 1500 pF/2KV or 1000 pF/3KV. A minimum of 50-mil spacing from capacitor to traces and components should be maintained.

14.3.1.2.1 Magnetics Module Qualification Steps

The steps involved in magnetics module qualification are similar to those for crystal qualification:

1. Verify that the vendor's published specifications in the component datasheet meet or exceed the required IEEE specifications.
2. Independently measure the component's electrical parameters on the test bench, checking samples from multiple lots. Check that the measured behavior is consistent from sample to sample and that measurements meet the published specifications.
3. Perform physical layer conformance testing and EMC (FCC and EN) testing in real systems. Vary temperature and voltage while performing system level tests.

14.3.1.2.2 Magnetics Module for 1000 BASE-T Ethernet

Magnetics modules for 1000 BASE-T Ethernet are similar to those designed solely for 10/100 Mbps, except that there are four differential signal pairs instead of two. Use the following guidelines to verify specific electrical parameters:

1. Verify that the rated return loss is 19 dB or greater from 2 MHz through 40 MHz for 100/1000 BASE-TX.
2. Verify that the rated return loss is 12 dB or greater at 80 MHz for 100 BASE-TX (the specification requires greater than or equal to 10 dB).
3. Verify that the rated return loss is 10 dB or greater at 100 MHz for 1000 BASE-TX (the specification requires greater than or equal to 8 dB).
4. Verify that the insertion loss is less than 1.0 dB at 100 kHz through 80 MHz for 100 BASE-TX.
5. Verify that the insertion loss is less than 1.4 dB at 100 kHz through 100 MHz for 1000 BASE-T.
6. Verify at least 30 dB of crosstalk isolation between adjacent channels (through 150 MHz).
7. Verify high voltage isolation to 15000 Vrms. (Does not apply to discrete magnetics.)
8. Transmitter OCL should be greater than or equal to 350 μH with 8 mA DC bias.

14.3.1.2.3 Third-Party Magnetics Manufacturers

The following magnetics modules have been used successfully in previous designs.

Table 14-2. Magnetics Manufacturers

Manufacturer	Part Number
Pulse	H5007
Bel (discrete)	Bel 0344FLA

14.3.1.2.4 Layout Guidelines for Use with Integrated and Discrete Magnetics

Layout requirements are slightly different when using discrete magnetics.

These include:

- ground cut for HV installation (not required for integrated magnetics)
- maximum of two (2) vias
- turns less than 45°
- discrete terminators

14.3.2 Designing with the 82576 GbE Controller

This section provides guidelines specific to the 82576 GbE controller.

14.3.2.1 LAN Disable for 82576 GbE Controller

The 82576 GbE Controller device has three signals that can be used for disabling Ethernet functions from system BIOS. LAN0_DIS_N and LAN1_DIS_N are the separate port disable signals and DEV_OFF_N is the device disable signal. Each signal can be driven from a system output port. Choose outputs from devices that retain their values during reset. For example, ICH7 resumes GPIO outputs (GP24, 25, 27, 28) transition high during reset. It is important not to use these signals to drive LAN0_DIS_N or LAN1_DIS_N because these inputs are latched upon the rising edge of PE_RST_N or an inband reset end. The DEV_OFF_N input is completely asynchronous and does not have this restriction.

For full details on the operation on the LAN Disable options in the 82576 GbE controller refer to section Section 4.2.3.

For full details on the operation on the Device Disable options in the 82576 GbE controller refer to section Section 4.4.

14.3.2.2 Serial EEPROM

The 82576 GbE Controller Gigabit Ethernet Controller uses an Serial Peripheral Interface (SPI)* EEPROM. Several words of the EEPROM are accessed automatically by the device after reset to provide pre-boot configuration data before it is accessed by host software. The remainder of the EEPROM space is available to software for storing the MAC address, serial numbers, and additional information. This information is available to the 82576 GbE Controller also and is part of the pre-boot configuration data.

For Full information on the EEPROM interface and operation refer to Section 3.3.1.

14.3.2.2.1 EEPROM-less Operation

The 82576 can be operated without an EEPROM, but conditions apply. Refer to Section 3.3.1.6.

14.3.2.2.2 SPI EEPROMs for 82576 GbE Controller

Atmel's AT25128N and Microchips 25LC128 Serial EEPROMs have been fully validated with the device and have been found to work satisfactorily. Alternate SPI EEPROMs that have been found to work with the 82576 are listed in Table 11-20. SPI EEPROMs must be rated for a clock rate of at least 2MHz. The recommended EEPROM size is 32k Byte (256k bit) EEPROM for all applications.

14.3.2.2.3 EEUPDATE

Intel has an MS-DOS* software utility called EEUPDATE. This utility can be used to program EEPROM images in development or production line environments. To obtain a copy, contact your Intel representative.

14.3.2.3 FLASH

For Full information on the EEPROM interface and operation refer to Section 3.3.4.

14.3.2.3.1 FLASH Device Information

While Intel does not make specific recommendations regarding FLASH devices, see Table 11-19 for a list of options.

14.3.3 SMBus and NC-SI

SMBus and NC-SI are optional interfaces for pass-through and configuration traffic between the BMC and the 82576 GbE Controller.

Note: Intel recommends that the SMBus be connected to the ICH or BMC for the EEPROM recovery solution. If the connection is to a BMC, it will be able to send the EEPROM release command.

The 82576 GbE Controller can be connected to an external BMC. It operates in one of two modes:

- SMBus mode
- NC-SI mode

The Clock-out (if enabled) is provided in all power states (unless the device is disabled).

For more information on system management solutions, see Chapter 10..

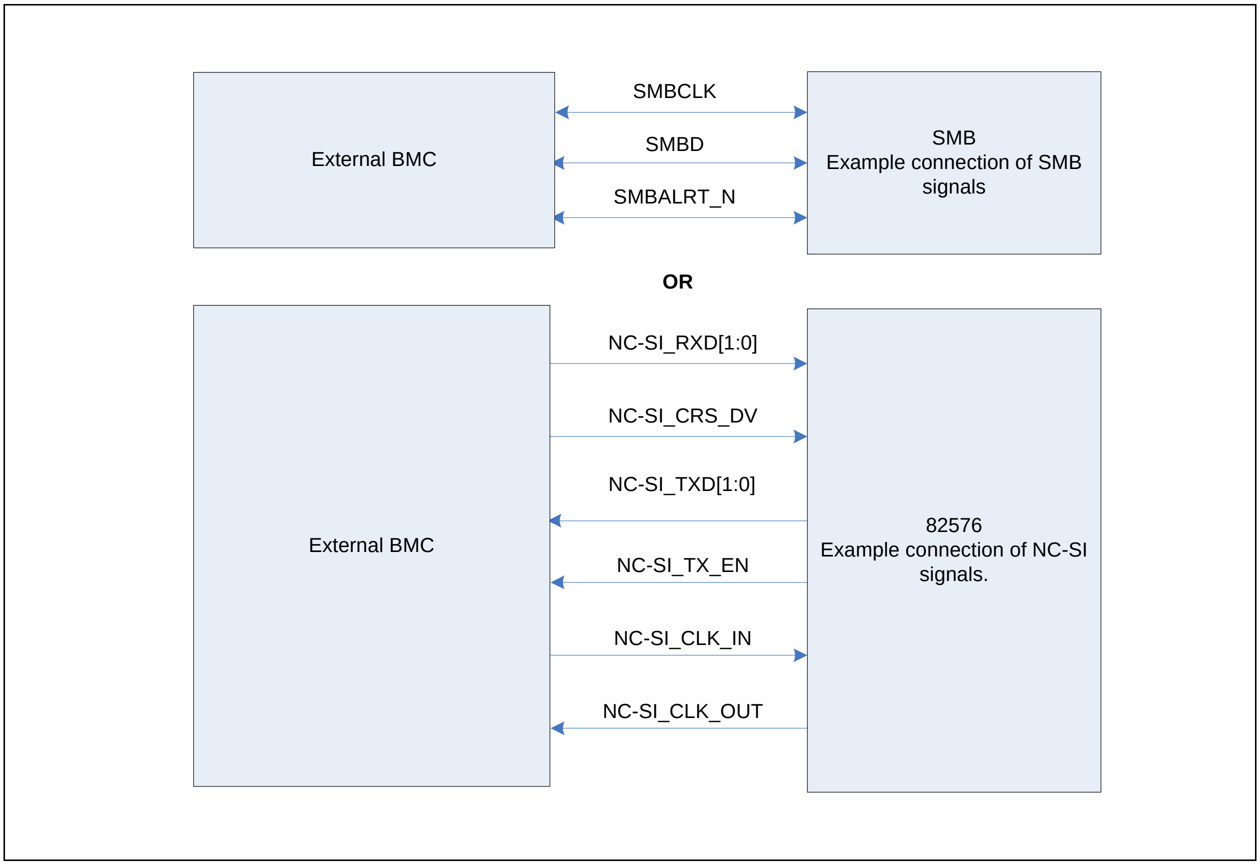


Figure 14-2. External BMC Connections with NC-SI and SMB

14.3.4 NC-SI Electrical Interface Requirements

This section describes the hardware implementation requirements necessary to meet the NC-SI physical layer standard. Board-level design requirements are included for connecting the 82576 Ethernet solution to an external Baseboard Management Controller (BMC). The layout and connectivity requirements are addressed in low-level detail. This section, in conjunction with the *Network Controller Sideband Interface Specification Version 0.8 NC-SI Specification*, also provides the complete board-level requirements for the NC-SI solution.

The 82576's on-board System Management Bus (SMBus) port enables network manageability implementations required for remote control and alerting via the LAN. With SMBus, management packets can be routed to or from an MC. Enhanced pass-through capabilities also enable system remote control over standardized interfaces. Also included is a new manageability interface, NC-SI that supports the DMTF preOS sideband protocol. An internal management interface called MDIO enables the MAC (and software) to monitor and control the PHY.

14.3.4.1 External Baseboard Management Controller (BMC)

The external BMC is required to meet the latest NC-SI specification as it relates to the RMII electrical interface.

14.3.4.2 Reference Schematic

Figure 14-3 shows the recommended pull-up and pull-downs to be always be used on the NC-SI interface regardless of the application, even when not using the interface.

- NCSI_CLK_IN: Series termination at the BMC and the 82576 should be used when using the interface. A pull down should always be placed on this input to the 82576.
- NCSI_CRS_DV: A pull-down should always be used to ensure the data valid is never indicated during start-up when the signal is not driven.
- NCSI_RXD_0/1: Pull-up resistors should always be used to ensure these lines at ‘1’ when nothing is driven or interface is not used and for multiple drop configurations.
- NCSI_TX_EN: A pull-down should always be used to ensure the TX is never enabled during start-up or when the signal is not driven.
- NCSI_TXD_0/1: Pull-up resistors should always be used to ensure these lines at ‘1’ when nothing is driven or interface is not used and for multiple drop configurations.

When using the NC-SI interface is used the BMC typically requires series termination close to these transmitters.

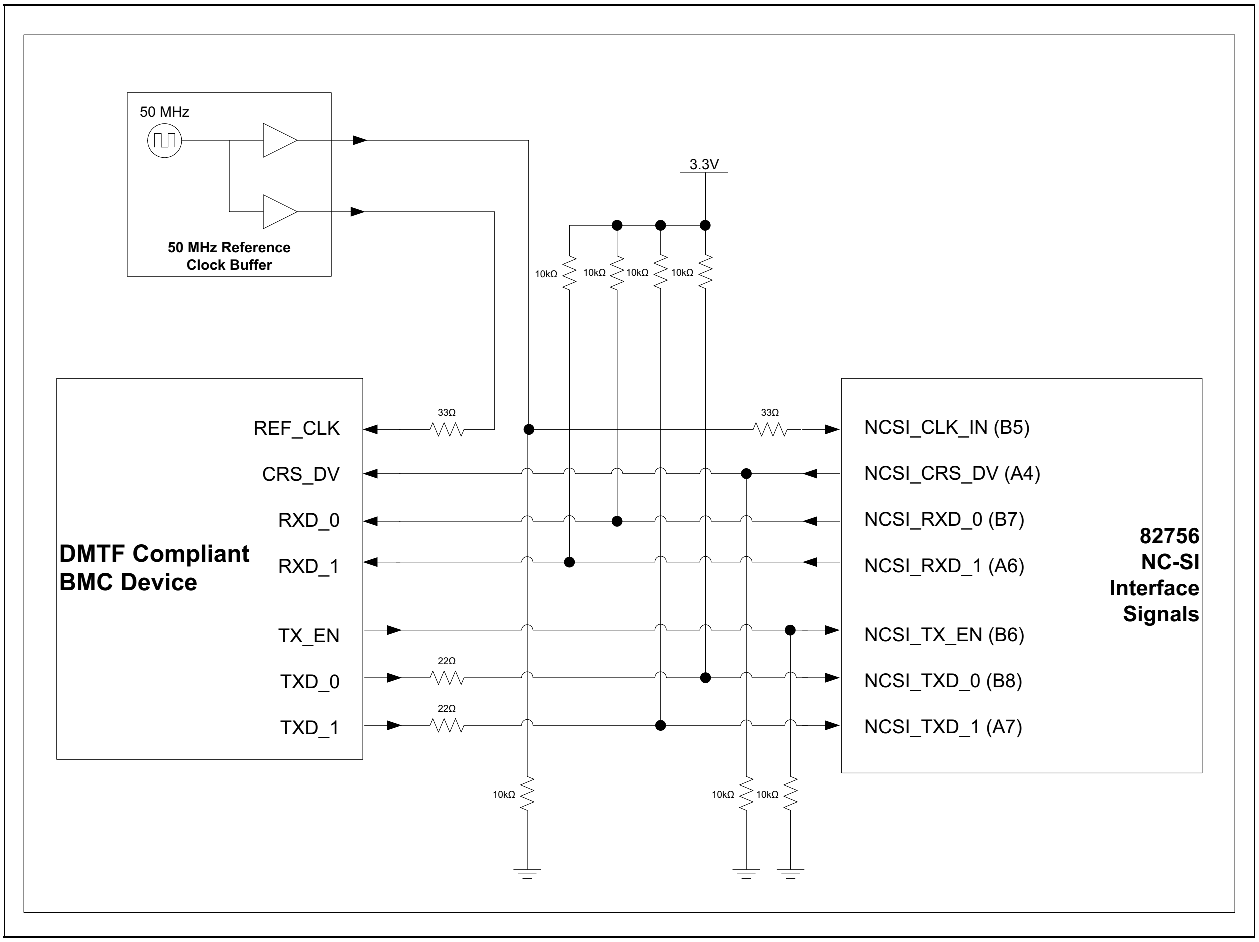


Figure 14-3. NC-SI Connection Requirement - Single Drop Configuration

14.3.4.3 Resets

It is important to ensure that the resets for the BMC and the 82576 are generated within a specific time interval. The important requirement here is ensuring that the NC-SI link is established within two seconds of the BMC receiving the power good signal from the platform. Both the 82576 and the external BMC need to receive power good signals from the platform within one second of each other.

This causes an internal power on reset within the 82576 and then initialization as well as a triggering and initialization sequence for the BMC. Once these power good signals are received by both the 82576 and the external BMC, the NC-SI interface can be initialized. The NC-SI specification calls out a requirement of link establishment within two seconds. The BMC should poll this interface and establish a link for two seconds to ensure specification compliance.

14.3.4.4 Layout Requirements

14.3.4.4.1 Board Impedance

The NC-SI signaling interface is a single-ended signaling environment with a target board and trace impedance of 50 Ω; plus 20% and minus 10% is recommended. This target impedance ensures optimal signal integrity and signal quality.

14.3.4.4.2 Trace Length Restrictions

Intel recommends a trace length maximum value from a board placement and routing topology perspective of eight inches for direct connect applications (Figure 14-4). This ensures that signal integrity and quality is preserved from a design perspective and that compliance is met for the NC-SI electrical requirements.

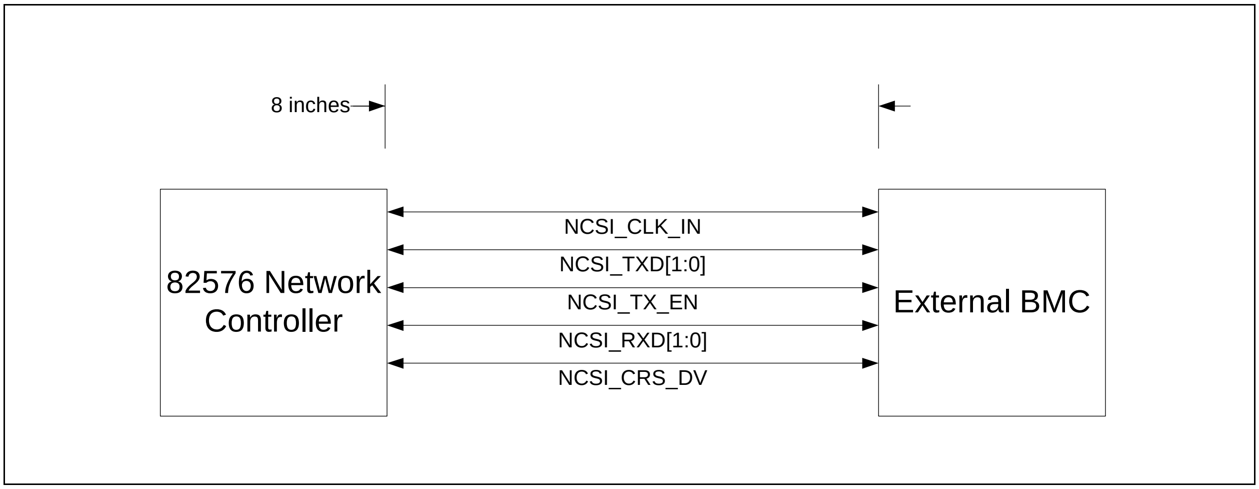


Figure 14-4. NC-SI Trace Length Requirement for Direct Connect

For multi-drop applications (Figure 14-5) the spacing recommendation is a maximum of four inches. This keeps the overall length between the BMC and the 82576 within the specification.

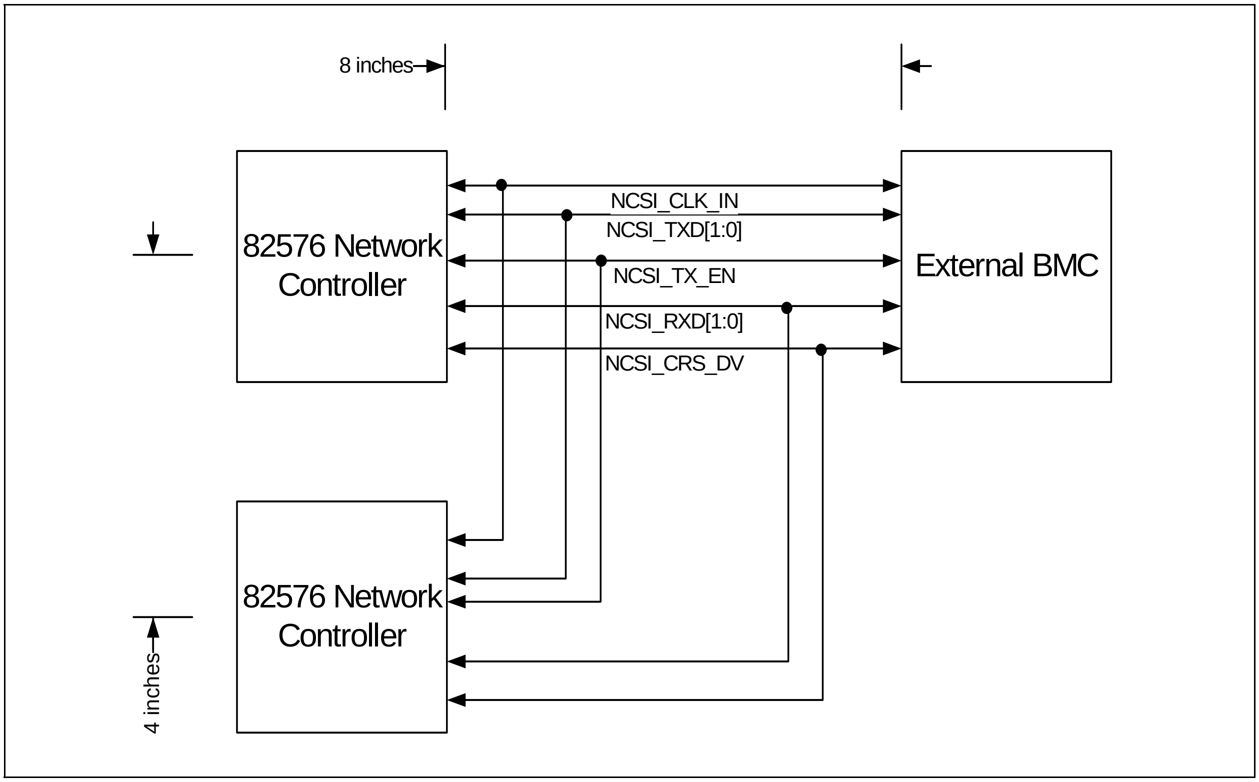


Figure 14-5. NC-SI Trace Length Requirement for Multi-Drop

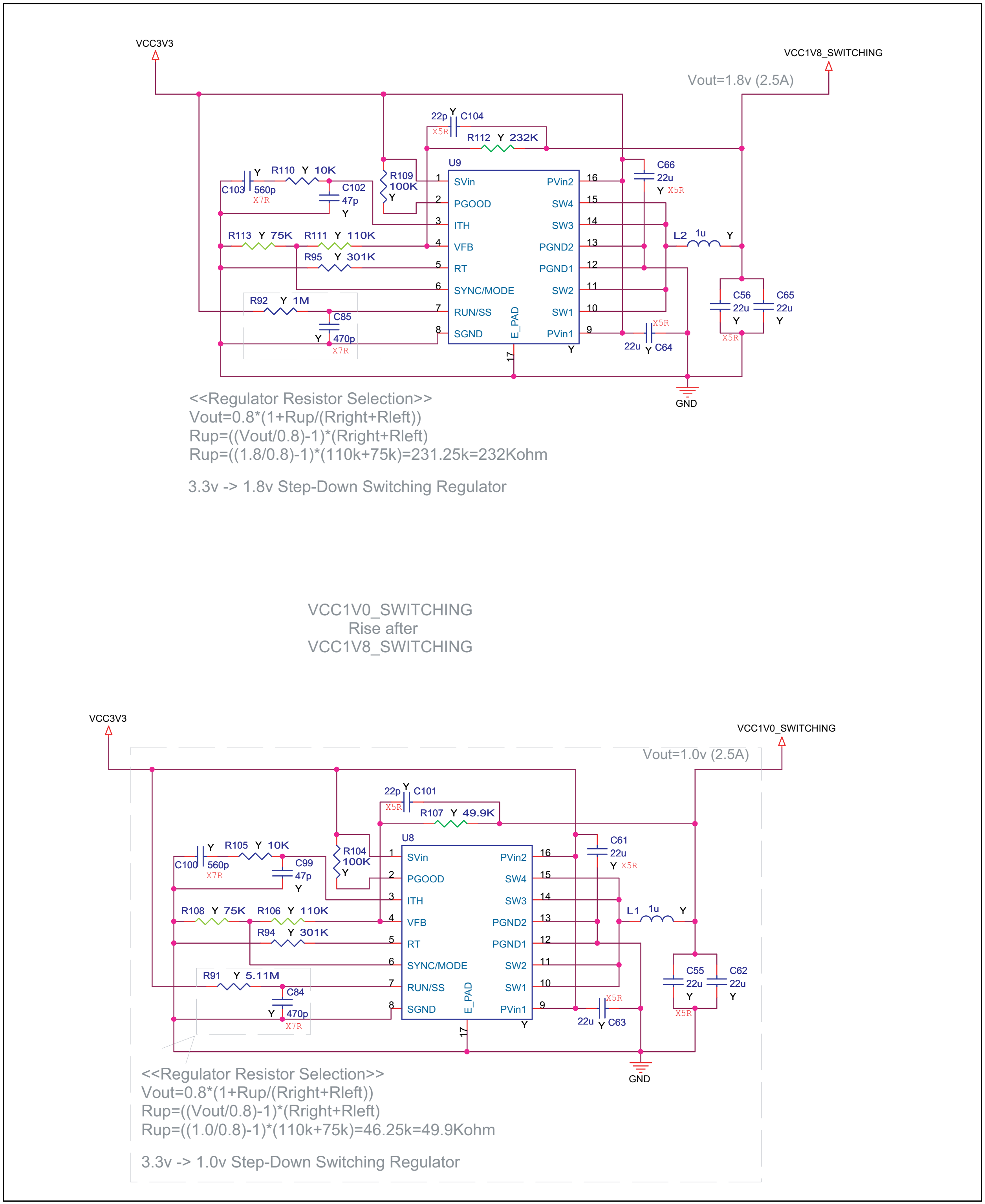
14.3.5 Power Supplies for the 82576 GbE Controller

The Intel® 82576 GbE Controller Gigabit Ethernet Controllers require three power rails: 3.3 V, 1.8 V and 1.0 V. A central power supply can provide all the required voltage sources, or the power can be derived from the 3.3 V supply and regulated locally using external regulators. If the LAN wake capability will be used, all voltages must remain present during system power down. Local regulation of the LAN voltages from system 3.3 Vmain and 3.3 Vaux voltages is recommended.

External voltage regulators need to generate the proper voltage, supply current requirements (with adequate margin), and provide the proper power sequencing.

Due to the current demand, a Switching Voltage Regulator (SVR) is highly recommended for the 1.0 V power rail. Figure 14-6 shows an example of a compact, low-part count, SVR that can be used for both the 1.0 V and 1.8 V power supplies.

The 1.8 V rail has a lower current requirement; however, the use of a SVR is still recommended for adequate margin. Using an LVR in this application is acceptable as long as adequate margin exists in the design and sequencing can be controlled. Figure 14-7 shows an example of a compact low-part -count LVR that could be used for the 1.8 V supply.



<<Regulator Resistor Selection>>
Vout=0.8*(1+Rup/(Rright+Rleft))
Rup=((Vout/0.8)-1)*(Rright+Rleft)
Rup=((1.8/0.8)-1)*(110k+75k)=231.25k=232Kohm

3.3v -> 1.8v Step-Down Switching Regulator

<<Regulator Resistor Selection>>
Vout=0.8*(1+Rup/(Rright+Rleft))
Rup=((Vout/0.8)-1)*(Rright+Rleft)
Rup=((1.0/0.8)-1)*(110k+75k)=46.25k=49.9Kohm

3.3v -> 1.0v Step-Down Switching Regulator

Figure 14-6. Example Switching Voltage Regulator for 1.0 V and 1.8 V

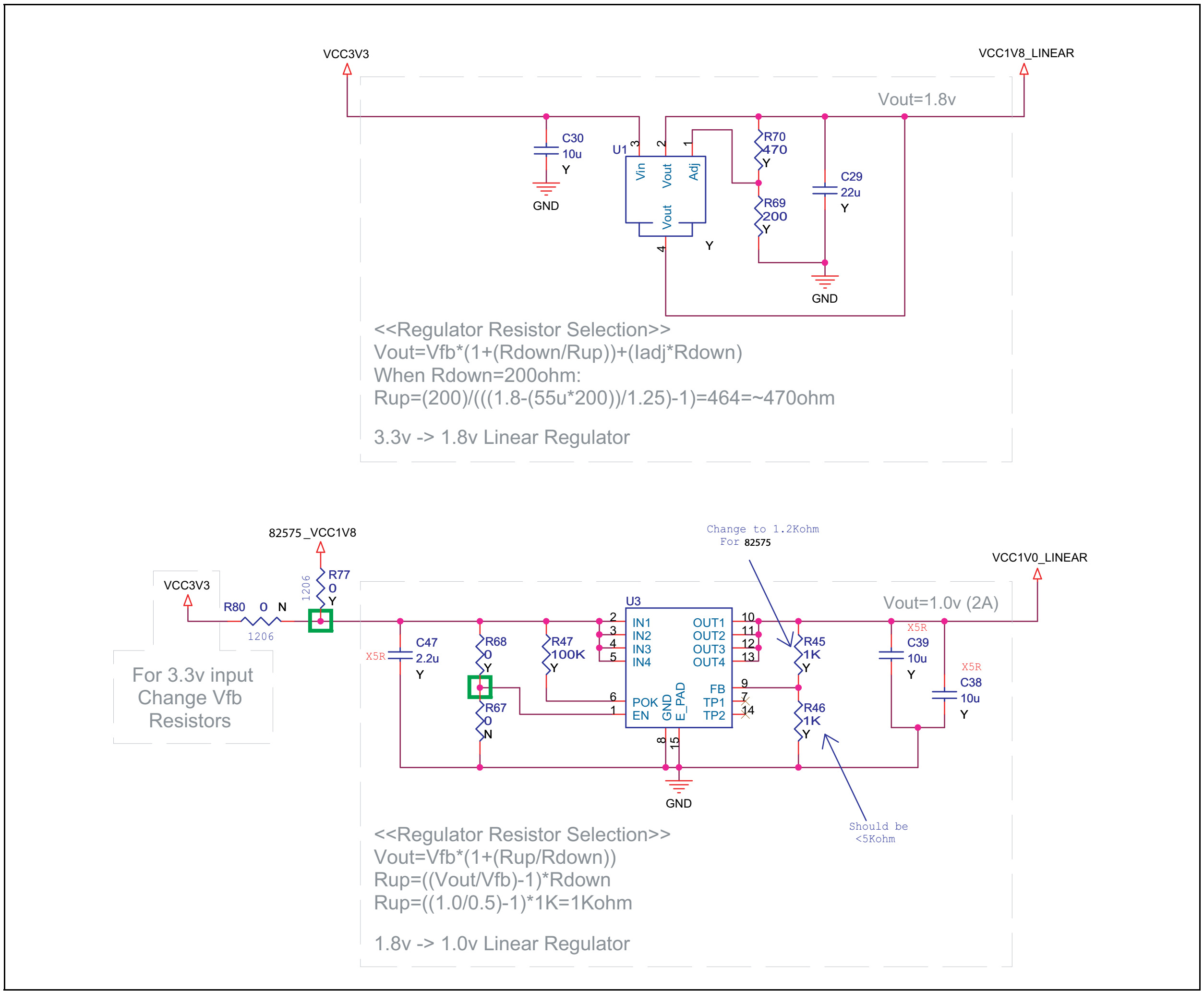


Figure 14-7. Example of Linear Voltage Regulator for 1.8 V power rail

14.3.5.1 82576 GbE Controller Power Sequencing

Regardless of which type of regulator used, all regulators need to adhere to the sequencing shown in Figure 14-8 to avoid latch-up and forward-biased internal diodes.

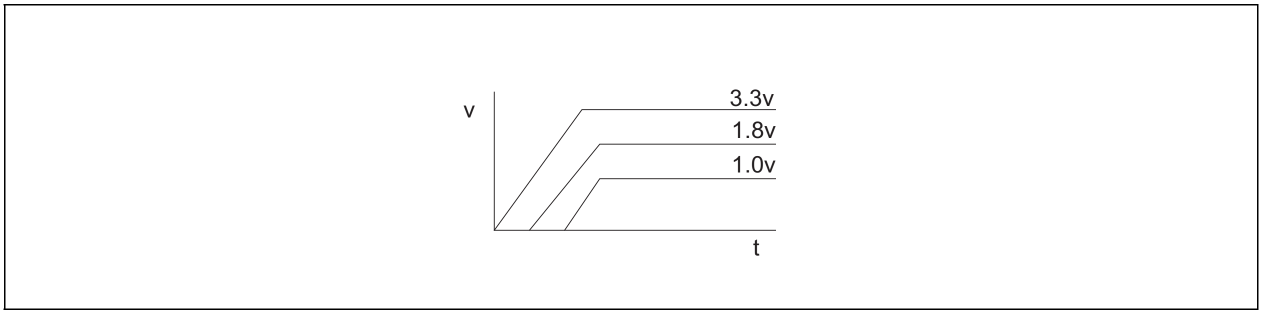


Figure 14-8. Proper power sequencing for 82576 GbE Controller

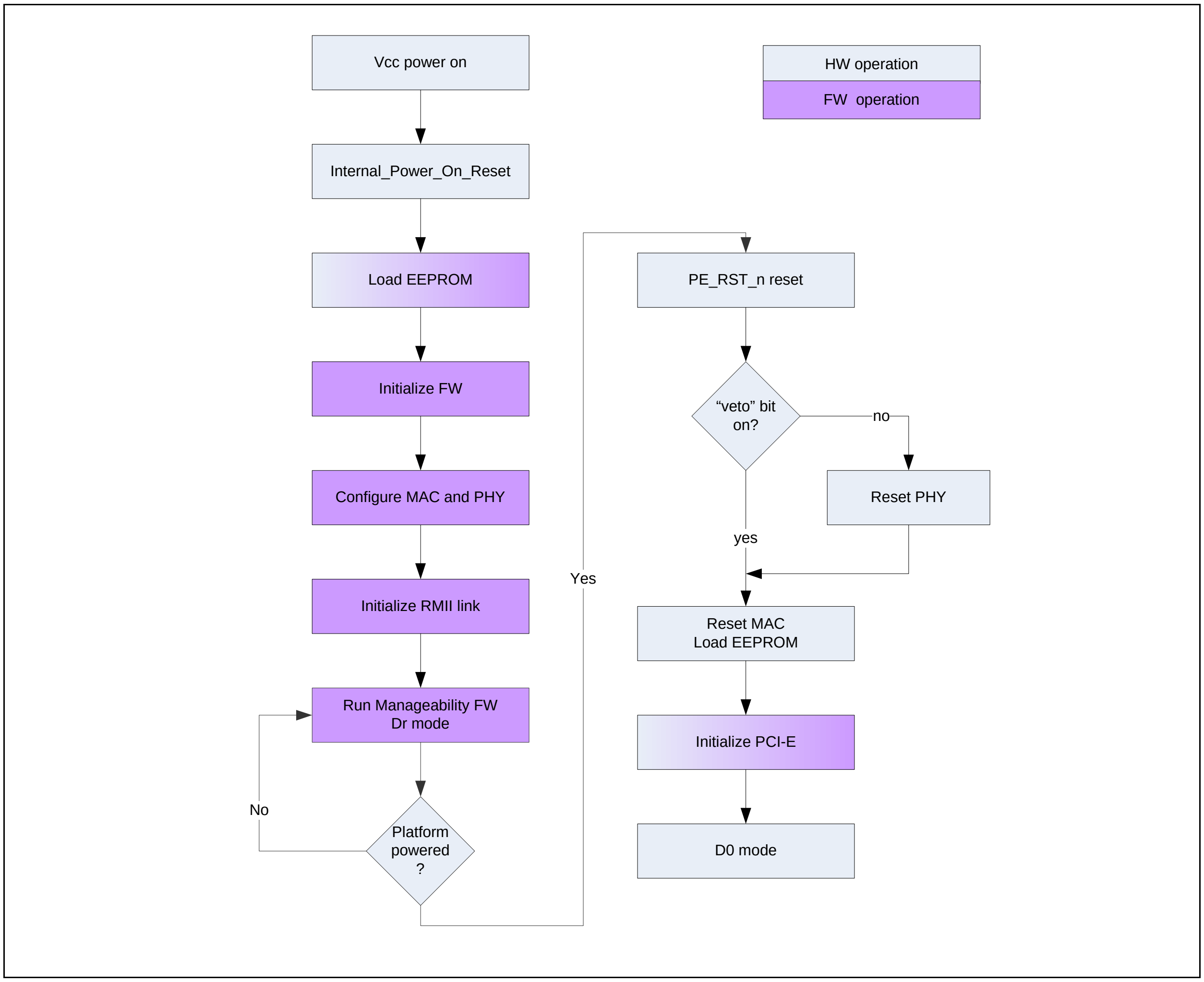


Figure 14-9. Power On Flowchart

In addition, the following limitations exist:

- 1.8 V must not exceed 3.3 V.
- 1.0 V must not exceed 3.3 V.
- 1.0 V must not exceed 1.8 V.

The power supplies are all expected to ramp during a short power-up internal (approximately 20ms or better). Do not leave the device in a prolonged state were some, but not all, voltages are applied.

14.3.5.1.1 Using Regulators With Enable Pins

The use of regulators with enable pins is very helpful in controlling sequencing. Connecting the enable of the 1.8 V regulator to 3.3 Vwill allow the 1.8 V to ramp. Connecting the enable of the 1.0 V regulator to the 1.8 V output assures that the 1.0 V rail will ramp after the 1.8 V rail. This provides a quick solution to power sequencing. Make sure to check design parameters for inputs with this configuration.

14.3.5.2 82576 GbE Controller Device Power Supply Filtering

Provide several high-frequency bypass capacitors for each power rail (see Table 14-3), selecting values in the range of 0.01µF to 0.1µF. If possible, orient the capacitors close to the device and adjacent to power pads. Decoupling capacitors should connect to the power planes with short, thick (18 mils or more) traces and 14 mil vias. Long and thin traces are more inductive and would reduce the intended effect of decoupling capacitors.

Table 14-3. Minimum Number of Bypass Capacitors per Power Rail

Power Rail	4.7uF or 10uF	0.1uF
3.3 V	1	2
1.8 V	1	4
1.0 V	1	6

Furnish approximately 4.7µF to 10µF of bulk capacitance for all the power rails; placement should be as close to the device power connection as possible.

14.3.5.3 82576 GbE Controller Power Management and Wake Up

The 82576 GbE Controller Gigabit Ethernet Controller supports low power operation as defined in the PCI Bus Power Management Specification. There are two defined power states, D0 and D3. The D0 state provides full power operation and is divided into two sub-states: D0u (uninitialized) and D0a (active). The D3 state provides low power operation and is also divided into two sub-states: D3hot and D3cold.

To enter the low power state (D3), the software driver must stop data transmission and reception. Either the operating system or the driver must program the Power Management Control/Status Register (PMCSR) and the Wakeup Control Register (WUC). If wakeup is desired, the appropriate wakeup LAN address filters must also be set. The initial power management settings are specified by EEPROM bits.

When the 82576 GbE Controller transitions to either of the D3 low power states, the 1.0 V, 1.8 V, and 3.3 V sources must continue to be supplied to the device. Otherwise, it will not be possible to use a wakeup mechanism. The AUX_PWR signal is a logic input to the 82576 GbE Controller that denotes auxiliary power is available. If AUX_PWR is asserted, the 82576 GbE Controller device will advertise that it supports wake up from a D3cold state.

The 82576 GbE Controller device supports both Advanced Power Management (APM) wakeup and Advanced Configuration and Power Interface (ACPI) wakeup. APM wakeup has also been known in the past as “Wake on LAN” and as “Magic Packet Wake-up”.

Wakeup uses the PE_WAKE_N signal to wake the system up. PE_WAKE_N is an active low signal typically connected to a GPIO port on the chipset that goes active in response to receiving a “Magic Packet”, a network wakeup packet, or link status change indication. PE_WAKE_N remains asserted until PME status is cleared in the 82576 Power Management Control/Status Register.

For more information on power management in the 82576, refer to Chapter 5..

14.3.6 82576 GbE Controller Device Test Capability

The 82576 GbE Controller Gigabit Ethernet Controller contains a test access port (3.3 V only) conforming to the IEEE 1149.1a-1994 (JTAG) Boundary Scan specification. To use the test access port, connect these balls to pads accessible by your test equipment.

A BSDL (Boundary Scan Definition Language) file describing the 82576 GbE Controller device is available for use in your test environment.

14.3.7 Software-Definable Pins (SDPs)

The Intel® 82576 GbE Controller has four software-defined pins (SDP) per port that can be used for hardware or software-programmable purposes. The pins are bound to a specific LAN device (eight SDPs may not be associated with a single LAN device). The pins can be individually configured to act as either input or output pins. The default direction is configurable EEPROM, as well as the default value of any pins configured as outputs.

Note: To avoid signal contention, the programmable pins are set as input pins until after the EEPROM configuration has been loaded.

For more information on the SDPs refer to Section 3.1.1.

14.4 Frequency Control Device Design Considerations

This section provides information regarding frequency control devices, including crystals and oscillators. Several suitable frequency control devices are available; none present any unusual challenges in selection. The concepts documented here are applicable to other data communication circuits, including PHYs.

Intel Ethernet controllers contain amplifiers, which when used with the specific external components, form the basis for feedback oscillators. Oscillator circuits, which are both economical and reliable, are described in detail in Section 14.5.

Intel Ethernet controllers also have bus clock input functionality. This functionality is not discussed in this document.

The chosen frequency control device vendor should be consulted early in the design cycle.

14.4.1 Frequency Control Component Types

Several types of frequency reference components are marketed. A discussion of each follows, listed in preferred order.

14.4.1.1 Quartz Crystal

Quartz crystals are the mainstay of frequency control components due to low cost and ease of implementation. They are available from numerous vendors in many package types with various specification options.

14.4.1.2 Fixed Crystal Oscillator

A packaged fixed crystal oscillator is comprised of an inverter, a quartz crystal, and passive components packaged together. The device renders a strong, consistent square wave output. Oscillators used with microprocessors are supplied in many configurations.

Crystal oscillators should be restricted to use in special situations, such as shared clocking among devices or multiple controllers. As clock routing can be difficult to accomplish, it is preferable to provide a separate crystal for each device.

For Intel Ethernet controllers, it is acceptable to overdrive the internal inverter by connecting a 25MHz external oscillator to the XTAL1 lead, leaving the XTAL2 lead unconnected. The oscillator should be specified to drive CMOS logic levels and the clock trace to the device should be as short as possible. Device specifications typically call for a 40% (minimum) to 60% (maximum) duty cycle and a ±50 ppm frequency tolerance.

Note: Please contact your Intel Customer Representative to obtain the most current device documentation prior to implementing this solution.

14.4.1.3 Programmable Crystal Oscillators

A programmable oscillator can be configured to operate at many frequencies. The device contains a crystal frequency reference and a phase lock loop (PLL) clock generator.

A programmable oscillator's accuracy depends heavily on the Ethernet device's differential transmit lines. The Physical Layer (PHY) uses the clock input from the device to drive a differential Manchester (for 10 Mbps operation), an MLT-3 (for 100 Mbps operation) or a PAM-5 (for 1000 Mbps operation) encoded analog signal across the twisted pair cable. These signals are referred to as self-clocking, which means the clock must be recovered at the receiving link partner. Clock recovery is performed with another PLL that locks onto the signal at the other end.

PLLs are prone to exhibit frequency jitter. The transmitted signal can also have considerable jitter even with the programmable oscillator working within its specified frequency tolerance. PLLs must be designed carefully to lock onto signals over a reasonable frequency range. If the transmitted signal has high jitter and the receiver's PLL loses its lock, then bit errors or link loss can occur.

PHY devices are deployed for many different communication applications. Some PHYs contain PLLs with marginal lock range and cannot tolerate the jitter inherent in data transmission clocked with a programmable oscillator. The American National Standards Institute (ANSI) X3.263-1995 standard test method for transmit jitter is not stringent enough to predict PLL-to-PLL lock failures, therefore, the use of programmable oscillators is generally not recommended.

14.4.1.4 Ceramic Resonator

Similar to a quartz crystal, a ceramic resonator is a piezoelectric device. A ceramic resonator typically carries a frequency tolerance of ±0.5%, inadequate for use with Intel Ethernet controllers and should not be utilized.

14.5 Crystal Selection Parameters

All crystals used with Intel Ethernet controllers are described as "AT-cut," which refers to the angle at which the unit is sliced with respect to the long axis of the quartz stone. Table 14-4 lists crystals which have been used successfully in other designs.

Table 14-4. Crystal Manufacturers and Part Numbers

Manufacturer	Part No.
RALTRON	AS-25.000-20-F-SMD-T
CITIZEN AMERICA CORP	HCM4925.000MBBKTR
NDK AMERICA INC	41CD25.0S11005020
TXC CORPORATION - USA	9C25000131

14.5.1 Vibrational Mode

Crystals in the above frequency range are available in both fundamental and third overtone. Unless there is a special need for third overtone, use fundamental mode crystals.

At any operating frequency, third overtone crystals are thicker and more rugged than fundamental mode crystals. Third overtone crystals are more suitable for use in military or harsh industrial environments. Third overtone crystals require a trap circuit (extra capacitor and inductor) in the load circuitry to suppress fundamental mode oscillation as the circuit powers up. Selecting values for these components is beyond the scope of this document.

14.5.2 Nominal Frequency

Intel Ethernet controllers use a crystal frequency of 25.000 MHz. The 25 MHz input is used to generate a 125 MHz transmit clock for 100BASE-TX and 1000BASE-TX operation; 10 MHz and 20 MHz transmit clocks for 10BASE-T operation.

14.5.3 Frequency Tolerance

The frequency tolerance for an Ethernet Platform LAN Connect is dictated by the IEEE 802.3 specification as ±50 parts per million (ppm). This measurement is referenced to a standard temperature of 25° C. Intel recommends a frequency tolerance of ±30 ppm.

14.5.4 Temperature Stability and Environmental Requirements

Temperature stability is a standard measure of how the oscillation frequency varies over the full operational temperature range (and beyond). Several optional temperature ranges are currently available, including -40° C to +85° C for industrial environments. Some vendors separate operating temperatures from temperature stability. Manufacturers may also list temperature stability as 50 ppm in their data sheets.

Note: Crystals also carry other specifications for storage temperature, shock resistance, and reflow solder conditions. Crystal vendors should be consulted early in the design cycle to discuss the application and its environmental requirements.

14.5.5 Calibration Mode

The terms “series-resonant” and “parallel-resonant” are used to describe crystal oscillator circuits. Specifying parallel mode is critical to determining how the crystal frequency is calibrated at the factory.

A crystal specified and tested as series resonant oscillates without problem in a parallel-resonant circuit, but the frequency is higher than nominal by several hundred parts per million. The purpose of adding load capacitors to a crystal oscillator circuit is to establish resonance at a frequency higher than the crystal’s inherent series resonant frequency.

Figure 14-10 illustrates a simplified schematic of the internal oscillator circuit. Pin X1 and X2 refers to XTAL1 and XTAL2 in the Ethernet device, respectively. The crystal and the capacitors form a feedback element for the internal inverting amplifier. This combination is called parallel-resonant, because it has positive reactance at the selected frequency. In other words, the crystal behaves like an inductor in a parallel LC circuit. Oscillators with piezoelectric feedback elements are also known as “Pierce” oscillators.

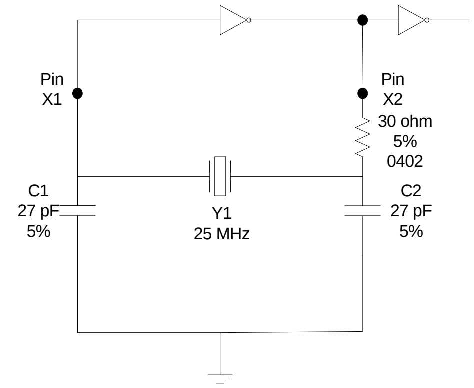


Figure 14-10. Internal Oscillator Circuit

14.5.6 Load Capacitance

The formula for crystal load capacitance is as follows:

$$C_L = \frac{(C1 \cdot C2)}{(C1 + C2)} + C_{stray}$$

where C1 = C2 = 27 pF

and C_{stray} = allowance for additional capacitance in pads, traces and the chip carrier within the Ethernet device package

An allowance of 3 pF to 7 pF accounts for lumped stray capacitance. The calculated load capacitance is 16 pF with an estimated stray capacitance of about 5 pF.

Individual stray capacitance components can be estimated and added. For example, surface mount pads for the load capacitors add approximately 2.5 pF in parallel to each capacitor. This technique is especially useful if Y1, C1 and C2 must be placed farther than approximately one-half (0.5) inch from the device. Thin circuit boards generally have higher stray capacitance than thick circuit boards. Consult the PCIe Design Guide for more information.

Oscillator frequency should be measured with a precision frequency counter where possible. The load specification or values of C1 and C2 should be fine tuned for the design. As the actual capacitance load increases, the oscillator frequency decreases.

Note: C1 and C2 may vary by as much as 5% (approximately 1 pF) from their nominal values.

14.5.7 Shunt Capacitance

The shunt capacitance parameter is relatively unimportant compared to load capacitance. Shunt capacitance represents the effect of the crystal's mechanical holder and contacts. The shunt capacitance should equal a maximum of 7 pF.

14.5.8 Equivalent Series Resistance

Equivalent Series Resistance (ESR) is the real component of the crystal's impedance at the calibration frequency, which the inverting amplifier's loop gain must overcome. ESR varies inversely with frequency for a given crystal family. The lower the ESR, the faster the crystal starts up. Use crystals with an ESR value of 50 Ω or better.

14.5.9 Drive Level

Drive level refers to power dissipation in use. The allowable drive level for a Surface Mounted Technology (SMT) crystal is less than its through-hole counterpart, because surface mount crystals are typically made from narrow, rectangular AT strips, rather than circular AT quartz blanks.

Some crystal data sheets list crystals with a maximum drive level of 1 mW. However, Intel Ethernet controllers drive crystals to a level less than the suggested 0.5 mW value. This parameter does not have much value for on-chip oscillator use.

14.5.10 Aging

Aging is a permanent change in frequency (and resistance) occurring over time. This parameter is most important in its first year because new crystals age faster than old crystals. Use crystals with a maximum of ±5 ppm per year aging.

14.5.11 Reference Crystal

The normal tolerances of the discrete crystal components can contribute to small frequency offsets with respect to the target center frequency. To minimize the risk of tolerance-caused frequency offsets causing a small percentage of production line units to be outside of the acceptable frequency range, it is important to account for those shifts while empirically determining the proper values for the discrete loading capacitors, C1 and C2.

Even with a perfect support circuit, most crystals will oscillate slightly higher or slightly lower than the exact center of the target frequency. Therefore, frequency measurements (which determine the correct value for C1 and C2) should be performed with an ideal reference crystal. When the capacitive load is exactly equal to the crystal's load rating, an ideal reference crystal will be perfectly centered at the desired target frequency.

14.5.11.1 Reference Crystal Selection

There are several methods available for choosing the appropriate reference crystal. These are listed below.

- If a Saunders and Associates (S&A) crystal network analyzer is available, then discrete crystal components can be tested until one is found with zero or nearly zero ppm deviation (with the appropriate capacitive load). A crystal with zero or near zero ppm deviation will be a good reference crystal to use in subsequent frequency tests to determine the best values for C1 and C2.
- If a crystal analyzer is not available, then the selection of a reference crystal can be done by measuring a statistically valid sample population of crystals, which has units from multiple lots and approved vendors. The crystal, which has an oscillation frequency closest to the center of the distribution, should be the reference crystal used during testing to determine the best values for C1 and C2.
- It may also be possible to ask the approved crystal vendors or manufacturers to provide a reference crystal with zero or nearly zero deviation from the specified frequency when it has the specified CLoad capacitance.

When choosing a crystal, keep in mind that to comply with IEEE specifications for 10/100 and 10/100/1000Base-T Ethernet LAN, the transmitter reference frequency must be precise within ±50 ppm. Intel® recommends using a transmitter reference frequency that is accurate to within ±30 ppm to account for variations in crystal accuracy due to crystal manufacturing tolerance.

14.5.11.2 Circuit Board

Since dielectric layers of the circuit board are allowed some reasonable variation in thickness, stray capacitance from the printed board (to the crystal circuit) will also vary. If the thickness tolerance for the outer layers of dielectric are controlled within ±17 percent of nominal, then the circuit board should not cause more than ±2 pF variation to the stray capacitance at the crystal. When tuning crystal frequency, it is recommended that at least three circuit boards are tested for frequency. These boards should be from different production lots of bare circuit boards.

Alternatively, a larger sample population of circuit boards can be used. A larger population will increase the probability of obtaining the full range of possible variations in dielectric thickness and the full range of variation in stray capacitance.

Next, the exact same crystal and discrete load capacitors (C1 and C2) must be soldered onto each board and the LAN reference frequency should be measured on each circuit board.

The circuit board, which has a LAN reference frequency closest to the center of the frequency distribution, should be used while performing the frequency measurements to select the appropriate value for C1 and C2.

14.5.11.3 Temperature Changes

Temperature changes can cause crystal frequency to shift. Frequency measurements should be done in the final system chassis across the system’s rated operating temperature range.

14.6 Oscillator Support

The Intel® 82576 GbE Controller clock input circuit is optimized for use with an external crystal. However, an oscillator can also be used in place of the crystal with the proper design considerations:

- The clock oscillator has an internal voltage regulator of 1.2 V to isolate it from the external noise, to minimize jitter. If an external clock is used, this imposes a maximum input clock amplitude of 1.2 V.
- The input capacitance introduced by the Intel® 82576 GbE Controller (approximately 20 pF) is greater than the capacitance specified by a typical oscillator (approximately 15 pF).
- The input clock jitter from the oscillator can impact the 82576 clock and its performance.

Table 14-5. Intel® 82576 GbE Controller Clock Oscillator Specifications

Symbol	Parameter	Specifications			Units
		Min	Typical	Max	
f0	Frequency (@25°C)	-	25	-	MHz
Vp-p	External Oscillator Supply Swing	3.0	3.3	3.6	V
Vscp-p	XTAL1 Swing	1.1	1.2	1.3	V
Df/f_o	Frequency Tolerance (@ -20 to +70 °C)	-	±50		ppm
Topr	Operating Temperature	-	-20 to +70		°C
$\Delta f/f_o$	Aging	-	±5		ppm/year
Ccoupling	Coupling Capacitor	8	10	12	pF

Note: The power consumption of additional circuitry equals about 1.5 mW.

Table 14-6 lists oscillators that have been used successfully in past designs (please note that no particular product is recommended):

Table 14-6. Oscillator Manufacturers and Part Numbers

Manufacturer	Part No.
RALTRON	CO4305-25.000-TR
CITIZEN AMERICA CORP	CSX750FBB25.000MTR

14.6.1 Oscillator Solution

This solution involves capacitor C1, which forms a capacitor divider with C_{stray} of about 20 pF. This attenuates the input clock amplitude and adjusts the clock oscillator load capacitance.

$$V_{in} = VDD * (C1/(C1 + C_{stray}))$$
$$V_{in} = 3.3 * (C1/(C1 + C_{stray}))$$

This enables load clock oscillators of 15 pF to be used. If the value of C_{stray} is unknown, C1 should be adjusted by tuning the input clock amplitude to approximately 1 V_{ptp}. If C_{stray} equals 20 pF, then C1 is 10 pF ±10%.

A low capacitance, high impedance probe (C < 1 pF, R > 500 KΩ) should be used for testing. Probing the parameters can affect the measurement of the clock amplitude and cause errors in the adjustment. A test should also be done after the probe has been removed for circuit operation.

If jitter performance is poor, a lower jitter clock oscillator can be implemented.

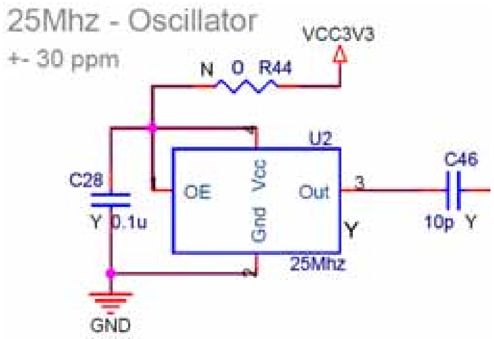


Figure 14-11. Reference Oscillator Circuit

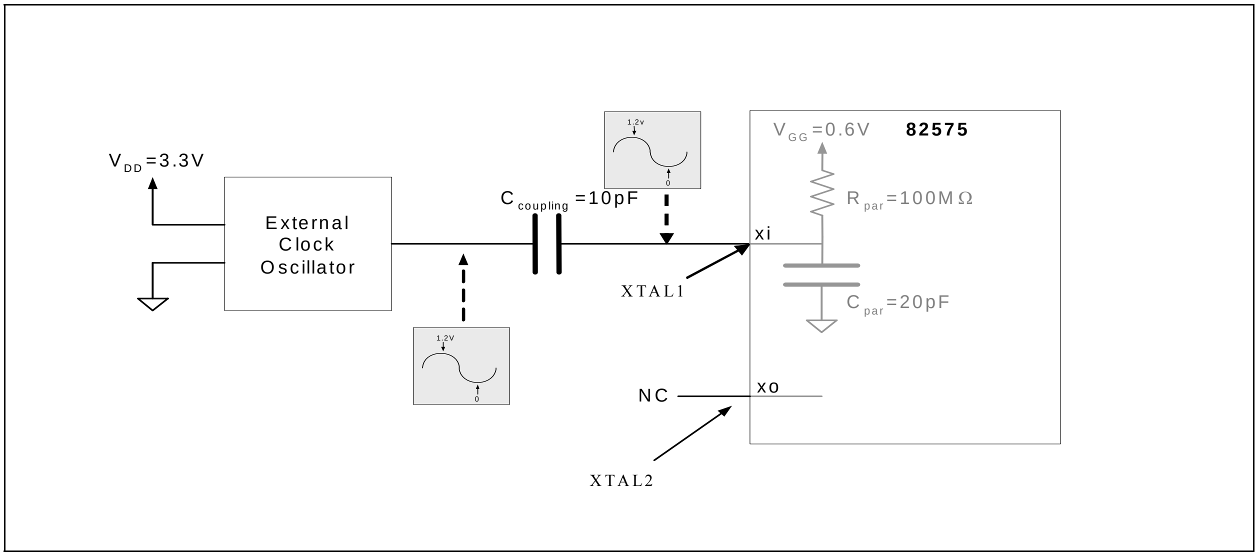


Figure 14-12. External Clock Oscillator Connectivity

14.7 Ethernet Component Layout Guidelines

These sections provide recommendations for performing printed circuit board layouts. Good layout practices are essential to meet IEEE PHY conformance specifications and EMI regulatory requirements.

14.7.1 Layout Considerations for Intel® 82576 GbE Controller Ethernet Controllers

Critical signal traces should be kept as short as possible to decrease the likelihood of being affected by high frequency noise from other signals, including noise carried on power and ground planes. Keeping the traces as short as possible can also reduce capacitive loading.

Since the transmission line medium extends onto the printed circuit board, special attention must be paid to layout and routing of the differential signal pairs.

Designing for 1000 BASE-T Gigabit operation is very similar to designing for 10 and 100 Mbps. For the 82576 Gigabit Ethernet controller, system level tests should be performed at all three speeds.

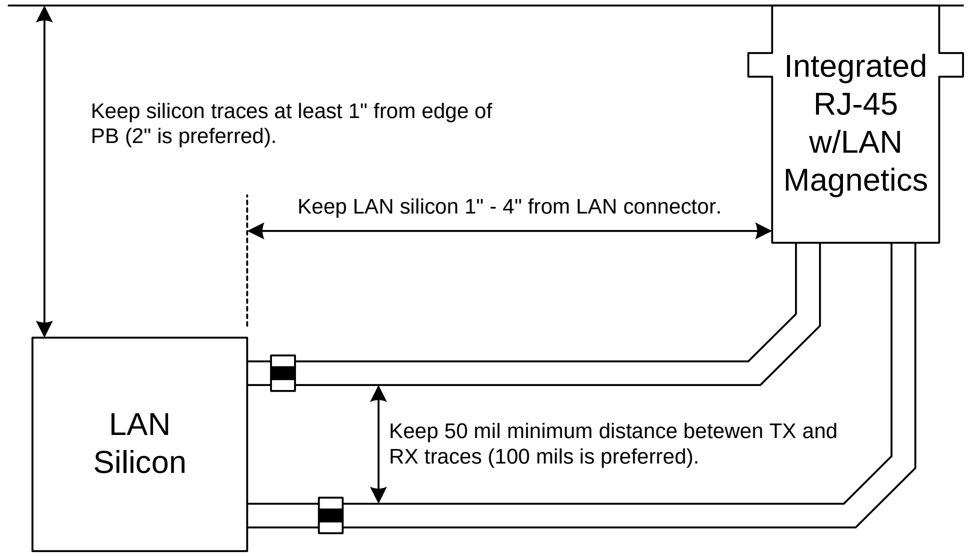
14.7.1.1 Guidelines for Component Placement

Component placement can affect signal quality, emissions, and component operating temperature This section provides guidelines for component placement. Careful component placement can:

- Decrease potential problems directly related to electromagnetic interference (EMI), which could cause failure to meet applicable government test specifications.
- Simplify the task of routing traces. To some extent, component orientation will affect the complexity of trace routing. The overall objective is to minimize turns and crossovers between traces.

Minimizing the amount of space needed for the Ethernet LAN interface is important because other interfaces will compete for physical space on a motherboard near the connector. The Ethernet LAN circuits need to be as close as possible to the connector.

Figure 14-13 shows some basic placement distance guidelines. Figure 14-13 shows two differential pairs, but can be generalized for a Gigabit system with four analog pairs. The ideal placement for the Ethernet silicon would be approximately one inch behind the magnetics module.


Keep silicon traces at least 1" from edge of PB (2" is preferred).
Integrated RJ-45 w/LAN Magnetics
Keep LAN silicon 1" - 4" from LAN connector.
LAN Silicon
Keep 50 mil minimum distance betewen TX and RX traces (100 mils is preferred).

Note: This figure represents a 10/100 diagram. Use the same design considerations for the two differential pairs not shown for gigabit implementations.

Figure 14-13. General Placement Distances for 1000 BASE-T Designs

While it is generally a good idea to minimize lengths and distances, Figure 14-13 also illustrates the need to keep the LAN silicon away from the edge of the board and the magnetics module for best EMI performance. Figure 14-14 and Figure 14-15 illustrate a reference layout for discrete and integrated magnetics.

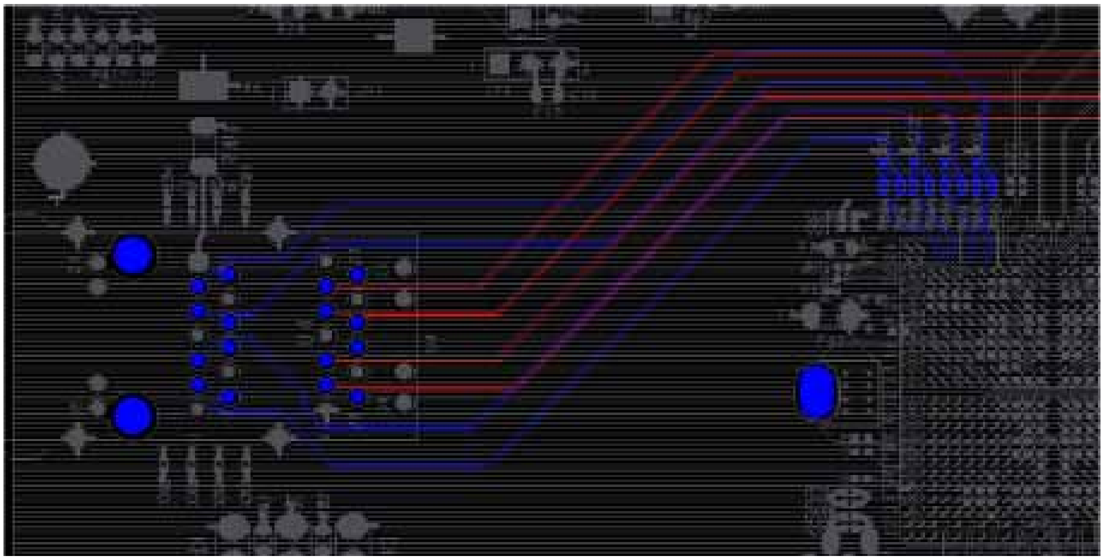


Figure 14-14. Layout for Integrated Magnetics

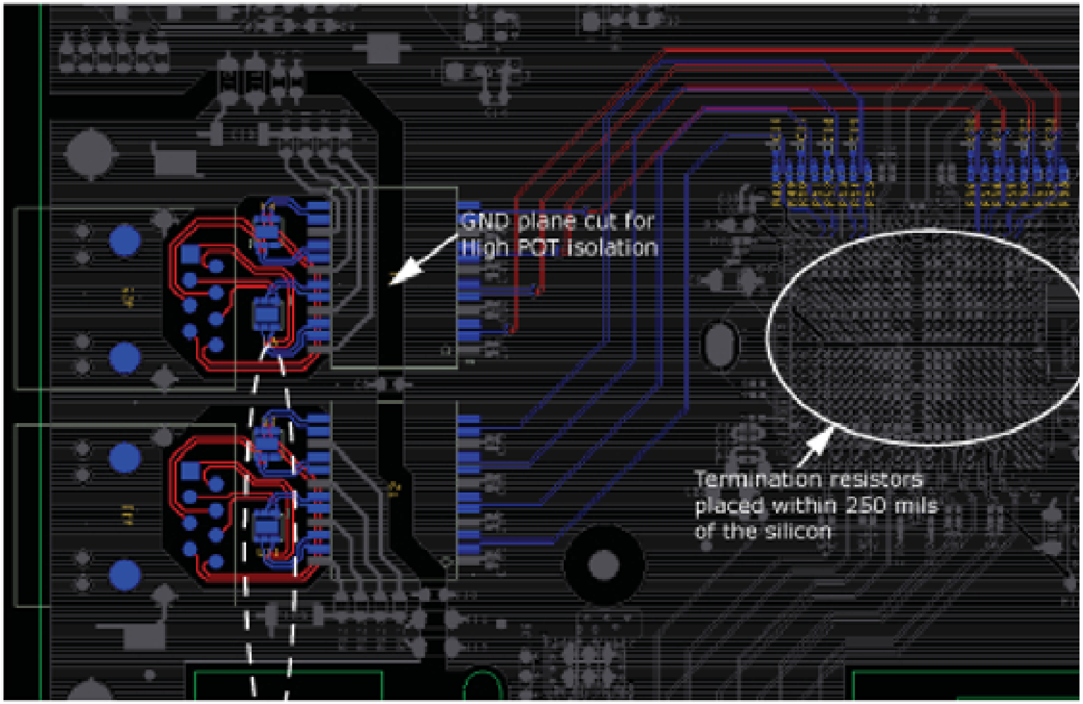


Figure 14-15. Layout for Discrete Magnetics

14.7.1.2 Crystals and Oscillators

Clock sources should not be placed near I/O ports or board edges. Radiation from these devices may be coupled into the I/O ports and radiate beyond the system chassis. Crystals should also be kept away from the Ethernet magnetics module to prevent interference.

14.7.1.2.1 Crystal layout considerations

Note: Failure to follow these guidelines could result in the 25 MHz clock failing to start.

When designing the layout for the crystal circuit, use the following rules:

- Place load capacitors as close as possible (within design-for-manufacturability rules) to crystal solder pads. They should be no more than 90 mils away from crystal pads.
- The two load capacitors, crystal component, the Ethernet controller device, and the crystal circuit traces must all be located on the same side of the circuit board (maximum of one via-to-ground load capacitor on each Xtal trace).
- Use 27 pF (5% tolerance) 0402 load capacitors.
- Place load capacitor solder pad directly in line with circuit trace (see Figure 14-16, point A).
- Place a 30-ohm (5% tolerance) 0402 series resistor on Xtal2 (see Figure 14-16, point C). The location of the resistor along the Xtal2 trace is flexible, as long as it is between the load capacitor and the controller.
- Use 50-ohm impedance single-ended microstrip traces for the crystal circuit.
- Route traces so that electro-magnetic fields from Xtal2 do not couple onto Xtal1. No differential traces.
- Route Xtal1 and Xtal2 traces to nearest inside corners of crystal pad (see Figure 14-16, point B).
- Ensure that the traces from Xtal1 and Xtal2 are symmetrically routed and that their lengths are matched.
- The total trace length of Xtal1 or Xtal2 should be less than 750 mils.

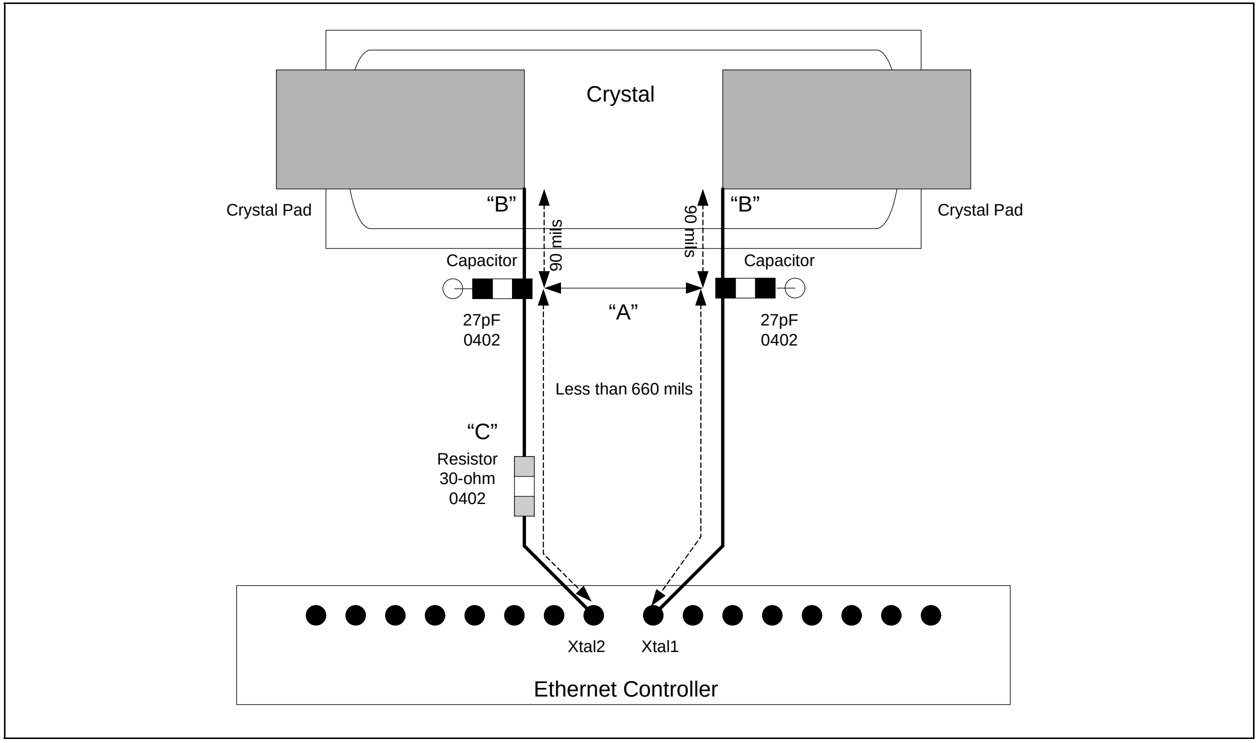


Figure 14-16. Recommended Crystal Placement and Layout

14.7.1.3 Board Stack Up Recommendations

Printed circuit boards for these designs typically have six, eight, or more layers. Although, the Intel® 82576 GbE Controller does not dictate the stackup, here is an example of a typical six-layer board stackup:

- Layer 1 is a signal layer. It can contain the differential analog pairs from the Ethernet device to the magnetics module, or to an optical transceiver.
- Layer 2 is a signal ground layer. Chassis ground may also be fabricated in Layer 2 under the connector side of the magnetics module.
- Layer 3 is used for power planes.
- Layer 4 is a signal layer.
- Layer 5 is an additional ground layer.
- Layer 6 is a signal layer. For 1000 BASE-T (copper) Gigabit designs, it is common to route two of the differential pairs (per port) on this layer.

This configuration can be adjusted to conform to your company's rules

14.7.1.4 Differential Pair Trace Routing for 10/100/1000 Designs

Trace routing considerations are important to minimize the effects of crosstalk and propagation delays on sections of the board where high-speed signals exist. Signal traces should be kept as short as possible to decrease interference from other signals, including those propagated through power and ground planes. Observe the following suggestions to help optimize board performance:

- Maintain constant symmetry and spacing between the traces within a differential pair.

- Minimize the difference in signal trace lengths of a differential pair.
- Keep the total length of each differential pair under 4 inches. Although possible, designs with differential traces longer than 5 inches are much more likely to have degraded receive BER (Bit Error Rate) performance, IEEE PHY conformance failures, and/or excessive EMI (Electromagnetic Interference) radiation.
- Do not route a pair of differential traces closer than 100 mils to another differential pair.
- Do not route any other signal traces parallel to the differential traces, and closer than 100 mils to the differential traces (300 mils is recommended).
- Keep maximum separation within differential pairs to 7 mils.
- For high-speed signals, the number of corners and vias should be kept to a minimum. If a 90° bend is required, it is recommended to use two 45° bends instead. See Figure 14-17.

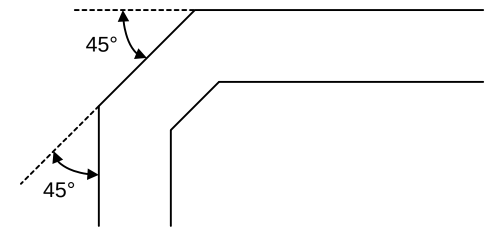


Figure 14-17. Trace Routing

Note: In manufacturing, vias are required for testing and troubleshooting purposes. The via size should be a 17-mil (±2 mils for manufacturing variance) finished hole size (FHS).

- Traces should be routed away from board edges by a distance greater than the trace height above the reference plane. This allows the field around the trace to couple more easily to the ground plane rather than to adjacent wires or boards.
- Do not route traces and vias under crystals or oscillators. This will prevent coupling to or from the clock. And as a general rule, place traces from clocks and drives at a minimum distance from apertures by a distance that is greater than the largest aperture dimension.
- The reference plane for the differential pairs should be continuous and low impedance. It is recommended that the reference plane be either ground or 1.8 V (the voltage used by the PHY). This provides an adequate return path for and high frequency noise currents.
- Do not route differential pairs over splits in the associated reference plane as it may cause discontinuity in impedances.

14.7.1.4.1 Signal Termination and Coupling

The four differential pairs of each port are terminated with 49.9 Ω (1% tolerance) resistors, placed near the controller. One resistor connects to the MDI+ signal trace and another resistor connects to the MDI- signal trace. The opposite ends of the resistors connect together and to ground through a single 0.1µF capacitor. The capacitor should be placed as close as possible to the 49.9 ohm resistors, using a wide trace. Stubs created by the 49.9 Ω (1% tolerance) termination resistors should be kept at a minimum.

Do not vary the suggested component values. Be sure to lay out symmetrical pads and traces for these components such that the length and symmetry of the differential pairs are not disturbed.

14.7.1.5 Signal Trace Geometry for 1000 BASE-T Designs

Key factors in controlling trace EMI radiation are the trace length and the ratio of trace-width to trace-height above the reference plane. To minimize trace inductance, high-speed signals and signal layers that are close to a reference or power plane should be as short and wide as practical. Ideally, this trace width to height above the ground plane ratio is between 1:1 and 3:1. To maintain trace impedance, the width of the trace should be modified when changing from one board layer to another if the two layers are not equidistant from the neighboring planes.

Each pair of signals should have a differential impedance of 100 Ω. +/- 15%. If a particular tool cannot design differential traces, it is permissible to specify 55-65 Ω single-ended traces as long as the spacing between the two traces is minimized. As an example, consider a differential trace pair on Layer 1 that is 8 mils (0.2 mm) wide and 2 mils (0.05 mm) thick, with a spacing of 8 mils (0.2 mm). If the fiberglass layer is 8 mils (0.2mm) thick with a dielectric constant, E_R, of 4.7, the calculated single-ended impedance would be approximately 61 Ω and the calculated differential impedance would be approximately 100 Ω.

When performing a board layout, do not allow the CAD tool auto-router to route the differential pairs without intervention. In most cases, the differential pairs will have to be routed manually.

Note: Measuring trace impedance for layout designs targeting 100 Ω often results in lower actual impedance. Designers should verify actual trace impedance and adjust the layout accordingly. If actual impedance is consistently low, a target of 105 – 110 Ω should compensate for second order effects.

It is necessary to compensate for trace-to-trace edge coupling (which can lower the differential impedance by up to 10 Ω) when the traces within a pair are closer than 30 mils (edge to edge).

14.7.1.6 Trace Length and Symmetry for 1000 BASE-T Designs

As indicated, the overall length of differential pairs should be less than four inches measured from the Ethernet device to the magnetics.

The differential traces (within each pair) should be equal in total length to within 50 mils (1.25mm) and as symmetrical as possible. Asymmetrical and unequal length traces in the differential pairs contribute to common mode noise. If a choice has to be made between matching lengths and fixing symmetry, more emphasis should be placed on fixing symmetry. Common mode noise can degrade the receive circuit’s performance and contribute to radiated emissions.

14.7.1.6.1 Signal Detect

Each port of the Intel® 82576 GbE Controller controller has a Signal Detect pin for connection to optical transceivers. For designs without optical transceivers, these signals can be left unconnected because they have internal pull-up resistors. Signal Detect is not a high-speed signal and does not require special layout.

14.7.1.7 Routing 1.8 V to the Magnetics Center Tap

The central-tap 1.8 V should be delivered as a solid supply plane (1.8 V) directly to the magnetic module or, if this is not possible, by a short and thick trace (lower than 0.2ohm DC resistance). The decoupling capacitors for the central tap pins should be placed as close as possible to the magnetic component. This improves both EMI and IEEE compliance.

14.7.1.8 Impedance Discontinuities

Impedance discontinuities cause unwanted signal reflections. Minimize vias (signal through holes) and other transmission line irregularities. If vias must be used, a reasonable budget is two per differential trace. Unused pads and stub traces should also be avoided.

14.7.1.9 Reducing Circuit Inductance

Traces should be routed over a continuous reference plane with no interruptions. If there are vacant areas on a reference or power plane, the signal conductors should not cross the vacant area. This causes impedance mismatches and associated radiated noise levels. Noisy logic grounds should be separated from analog signal grounds to reduce coupling. Noisy logic grounds can sometimes affect sensitive DC subsystems such as analog to digital conversion, operational amplifiers, etc.

All ground vias should be connected to every ground plane; similarly, every power via, to all power planes at equal potential. This helps reduce circuit inductance. Another recommendation is to physically locate grounds to minimize the loop area between a signal path and its return path. Rise and fall times should be as slow as possible (because signals with fast rise and fall times contain many high frequency harmonics, which can radiate significantly).

The most sensitive signal returns closest to the chassis ground should be connected together. This will result in a smaller loop area and reduce the likelihood of crosstalk. The effect of different configurations on the amount of crosstalk can be studied using electronics modeling software.

14.7.1.10 Signal Isolation

To maintain best signal integrity, keep digital signals far away from analog traces. A good rule of thumb is no digital signal should be within 300 mils (7.5mm) of differential pairs. If digital signals on other board layers cannot be separated by a ground plane, they should be routed perpendicular to differential pairs. If there is another LAN controller on the board, take care to keep differential pairs away from that circuit.

Rules follow for signal isolation:

- Separate and group signals by function on separate layers if possible. Maintain a gap of 100 mils between all differential pairs (Ethernet) and other nets, but group associated differential pairs together. Over the length of the trace run, each differential pair should be at least 0.3 inches away from any parallel signal traces.
- Physically group components associated with one clock trace to reduce trace length and radiation.
- Isolate I/O signals from high-speed signals to minimize crosstalk, which can increase EMI emission and susceptibility to EMI.
- Avoid routing high-speed LAN traces near other high-frequency signals associated with a video controller, cache controller, processor, or other similar devices.

14.7.1.11 Power and Ground Planes

Good grounding requires minimizing inductance levels in the interconnections and keeping ground returns short, signal loop areas small, and power inputs bypassed to signal return, will significantly reduce EMI radiation.

These guidelines help reduce circuit inductance in backplanes and motherboards:

- Route traces over a continuous plane with no interruptions. Do not route them over a split power or ground plane. If there are vacant areas on a ground or power plane, avoid routing signals over the vacant area. This will increase inductance and EMI radiation levels.

- Separate noisy digital grounds from analog grounds to reduce coupling. Noisy digital grounds may affect sensitive DC subsystems.
- All ground vias should be connected to every ground plane; every power via should be connected to all power planes at equal potential. This reduces circuit inductance.
- Physically locate grounds between a signal path and its return. This minimizes the loop area.
- Avoid fast rise/fall times as much as possible. Signals with fast rise and fall times contain many high frequency harmonics (which can radiate EMI).
- The ground plane beneath a magnetics module should be split. The RJ45 connector side of the transformer module should have a chassis ground beneath it.

14.7.1.12 Traces for Decoupling Capacitors

Traces between decoupling and I/O filter capacitors should be as short and wide as practical. Long and thin traces are more inductive and reduce the intended effect of decoupling capacitors. For similar reasons, traces to I/O signals and signal terminations should be as short as possible. Vias to the decoupling capacitors should be sufficiently large in diameter to decrease series inductance.

14.7.1.13 Light Emitting Diodes for Designs Based on the 82576 Controller

The Intel® 82576 GbE Controller provides four programmable high-current push-pull (active high) outputs per port to directly drive LEDs for link activity and speed indication. Each LAN device provides an independent set of LED outputs; these pins and their function are bound to a specific LAN device. Each of the four LED outputs can be individually configured to select the particular event, state, or activity, which will be 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.

Since the LEDs are likely to be integral to a magnetics module, take care to route the LED traces away from potential sources of EMI noise. In some cases, it may be desirable to attach filter capacitors.

The LED ports are fully programmable through the EEPROM interface.

14.7.1.14 Thermal Design Considerations

The Intel® 82576 GbE Controller contains a thermal sensor that is accessible through the SMBus. Trip points can be set in the EEPROM for the device. IcePak* and FlowTherm* models are available for the 82576 GbE Controller; contact your Intel representative for information.

14.7.2 Physical Layer Conformance Testing

Physical layer conformance testing (also known as IEEE testing) is a fundamental capability for all companies with Ethernet LAN products. PHY testing is the final determination that a layout has been performed successfully. If your company does not have the resources and equipment to perform these tests, consider contracting the tests to an outside facility.

14.7.2.1 Conformance Tests for 10/100/1000 Mbps Designs

Crucial tests are as follows, listed in priority order:

- Bit Error Rate (BER). Good indicator of real world network performance. Perform bit error rate testing with long and short cables and many link partners. The test limit is 10^{-11} errors.
- Output Amplitude, Rise and Fall Time (10/100Mbps), Symmetry and Droop (1000Mbps). For this controller, use the appropriate PHY test waveform.
- Return Loss. Indicator of proper impedance matching, measured through the RJ-45 connector back toward the magnetics module.

- Jitter Test (10/100Mbps) or Unfiltered Jitter Test (1000Mbps). Indicator of clock recovery ability (master and slave for Gigabit controller).

14.7.3 Troubleshooting Common Physical Layout Issues

The following is a list of common physical layer design and layout mistakes in LAN On Motherboard Designs.

1. Lack of symmetry between the two traces within a differential pair. Asymmetry can create common-mode noise and distort the waveforms. For each component and/or via that one trace encounters, the other trace should encounter the same component or a via at the same distance from the Ethernet silicon.
2. Unequal length of the two traces within a differential pair. Inequalities create common-mode noise and will distort the transmit or receive waveforms.
3. Excessive distance between the Ethernet silicon and the magnetics. Long traces on an FR4 fiberglass epoxy substrate will attenuate the analog signals. In addition, any impedance mismatch in the traces will be aggravated if they are longer than the four inch guideline.
4. Routing any other trace parallel to and close to one of the differential traces. Crosstalk getting onto the receive channel will cause degraded long cable BER. Crosstalk getting onto the transmit channel can cause excessive EMI emissions and can cause poor transmit BER on long cables. At a minimum, other signals should be kept 0.3 inches from the differential traces.
5. Routing one pair of differential traces too close to another pair of differential traces. After exiting the Ethernet silicon, the trace pairs should be kept 0.3 inches or more away from the other trace pairs. The only possible exceptions are in the vicinities where the traces enter or exit the magnetics, the RJ-45 connector, and the Ethernet silicon.
6. Use of a low-quality magnetics module.
7. Re-use of an out-of-date physical layer schematic in a Ethernet silicon design. The terminations and decoupling can be different from one PHY to another.
8. Incorrect differential trace impedances. It is important to have ~100 W impedance between the two traces within a differential pair. This becomes even more important as the differential traces become longer. To calculate differential impedance, many impedance calculators only multiply the single-ended impedance by two. This does not take into account edge-to-edge capacitive coupling between the two traces. When the two traces within a differential pair are kept close to each other, the edge coupling can lower the effective differential impedance by 5 W to 20 W. Short traces will have fewer problems if the differential impedance is slightly off target.

14.8 Serdes Implementation

This section clarifies Serdes implementation. See also: *Serdes Application Note AN498*.

14.8.0.1 Connecting the Serdes Interface

Table 14-7. Connecting the Serdes Interface

Signal Name	BX Backplane Connector		SFF(LASER) Transceiver		SFP Connector	
	Connection	PU/PD	Connection	PU/PD	Connection	PU/PD
SRDSI_0_p	x	--	RDM	--	RD+	--
SRDSI_0_n	x	--	RDP	--	RD-	--
SRDSO_0_p	x	--	TDP	--	TD+	--
SRDSO_0_n	x	--	TDM	--	TD+	--
SRDS_0_SIG_DET		PU	SD		--	PU
SDP0_0		--	--	--	--	--

Table 14-7. Connecting the Serdes Interface

SDP0_1 [1]	--	--	--		TXFAULT	PU
SDP0_2 [2]	--	--	TDIS	PD	TXDIS	PU
SDP0_3 [2]	--	--	LASER_PWR	--	LOS	PU
SFP0_I2C_CLK [3]	--	--	--	--	MOD-DEF1	PU
SFP0_I2C_DATA	--	--	--	--	MOD-DEF2	PU
SRDSI_1_p	x	--	RDM	--	RD+	--
SRDSI_1_n	x	--	RDP	--	RD-	--
SRDSO_1_p	x	--	TDP	--	TD+	--
SRDSO_1_n	x	--	TDM	--	TD+	--
SRDS_1_SIG_DET		PU	SD		--	PU
SDP1_0						
SDP1_1 [1]	--	--	--	--	TXFAULT	PU
SDP1_2 [1]	--	--	TDIS	PD	TXDIS	PU
SDP1_3 [1]	--	--	LASER_PWR	--	LOS	PU
SFP1_I2C_CLK [3]	--	--	--	--	MOD-DEF1	PU
SFP1_I2C_DATA	--	--	--	--	MOD-DEF2	PU

Notes:

1. SDP pins are software definable pins; however, use this implementation as it has been tested and verified with Intel drivers. Failure to follow these recommendations could cause interoperability issues.
2. PU value – 10K.
3. The SFPn_I2C_CLK does not allow for clock stretching.

14.8.0.2 Output voltage Adjustment

SerDes differential amplitude can be adjusted by EEPROM or Register modification.

- EEPROM – To obtain an EEPROM image that sets amplitude other then default, contact Intel.
- Using the Register – See 0x00024 SERDESSCTL.

14.8.0.3 Output Voltage Adjustment

Table 14-8. Output Voltage Adjustment

Reg. Number	0x00	0x34	Diff Amp [mV Pk-Pk]	
			Core 0	Core 1
Min Reg. Value	0x00	0x05	134	136
Min Amp -1step	0x8C	0x05	684	689
Min Amp ~750mV Pk-Pk	0x9C	0x05	741	750
Min Amp +1	0xAC	0x05	797	806
Default by EEPROM	0xFC	0x05	1.072	1.06
Default by EEPROM +1step	0x0C	0x15	1.131	1.117
MAX Reg. Value	0xFC	0x15	1.738	1.74

14.9 Reference Schematics

See Chapter 16., “Schematics.”

14.10 Thermal Management

See Chapter 15., “Thermal Design Specifications.”

14.11 Design and Layout Checklists

Checklists are available from your Intel representative.

14.12 Symbols

The CAD model for this product is available from your Intel representative.

§ §

NOTE: This page intentionally left blank.

15. Thermal Design Specifications

15.1 Product Package Thermal Specification

Table 15-1. Package Thermal Characteristics in a Standard System Environment

Package Type	Measured Power (Thermal Design Power*)	ΘJA	ΨJT
25mm 576FCBGA5-4L No thermal solution	2.4 W	18.4˚C/W	1.7˚C/W
25mm 576FCBGA5-4L With thermal solution	2.4 W	13.5˚C/W	0.6˚C/W

* See Section 15.5 for a definition of this parameter.

The thermal parameters defined above are for reference only and based on a combination of empirical/ simulated results of packages assembled on a standard 4s4p 1.0-oz Cu signal layer, 1.0-oz Cu power/ ground layer board in a natural convection environment. ΘJA is the package junction-to-air thermal resistance. ΨJT is the junction-to-package top thermal characterization parameter. System designs may vary considerably from the typical JEDEC board environment used.

Package thermal models are available upon request (Flotherm 2-Resistor, Delphi or Detailed and Icepak format).

15.2 Introduction

This chapter describes the thermal characteristics for Intel® 82576EA/EB/ES. Use this document to design a thermal solution for systems implementing the device.

Properly designed solutions provide adequate cooling to maintain the case temperature (Tcase) at or below those listed in Table 15-2. Ideally, this is accomplished by providing a low local ambient temperature and creating a minimal thermal resistance to that local ambient temperature. Heat sinks may be required if case temperatures exceed those listed. Operating outside of these operating limits may result in improper functionality or permanent damage to the Intel component and potentially other components within the system. Maintaining the proper thermal environment is essential to reliable, long-term component/system operation.

15.3 Measuring the Thermal Conditions

This chapter provides a method for determining the operating temperature of the device in a specific system based on case temperature. Case temperature is a function of the local ambient and internal temperatures of the component. This document specifies a maximum allowable Tcase for the device.

15.4 Thermal Considerations

In a system environment, the temperature of a component is a function of both the system and component thermal characteristics. System-level thermal constraints consist of the local ambient temperature at the component, the airflow over the component and surrounding board, and the physical constraints at, above, and surrounding the component that may limit the size of a thermal enhancement (heat sink).

The component’s case/die temperature depends on:

- component power dissipation
- size
- packaging materials (effective thermal conductivity)
- type of interconnection to the substrate and motherboard
- presence of a thermal cooling solution
- power density of the substrate, nearby components, and motherboard

These parameters are pushed by the continued trend of technology to increase performance levels (higher operating speeds, MHz) and power density (more transistors). As operating frequencies increase and packaging size decreases, the power density increases and the thermal cooling solution space and airflow become more constrained. The result is an increased emphasis on system design to ensure that thermal design requirements are met for each component in the system.

The thermal management objective is to ensure that all system component temperatures are maintained within functional limits. The functional temperature limit is the range in which the electrical circuits are expected to meet specified performance requirements. Operation outside the functional limit can degrade system performance, cause logic errors, or cause component and/or system damage. Temperatures exceeding the maximum operating limits may result in irreversible changes in the component operating characteristics. Also note that sustained operation at component maximum temperature limit may affect long-term device reliability.

15.5 Packaging Terminology

The following is a list of packaging terminology used in this document:

- **FCBGA Flip Chip Ball Grid Array**: A surface mount package using a combination of flip chip and BGA structure whose PCB-interconnect method consists of eutectic solder ball array on the interconnect side of the package. The die is flipped and connected to an organic build-up substrate with C4 bumps. An integrated heat spreader (IHS) may be present for larger FCBGA packages for enhanced thermal performance.
- **Junction**: Refers to a P-N junction on the silicon. In this document, it is used as a temperature reference point (for example, $\text{Theta}_{\_JA}$ refers to the “junction” to “ambient” thermal resistance).
- **Ambient**: Refers to local ambient temperature of the bulk air approaching the component. It can be measured by placing a thermocouple approximately 1” upstream from the component edge.
- **Lands**: The pads on the PCB to which BGA Balls are soldered.
- **PCB**: Printed Circuit Board.
- **Printed Circuit Assembly (PCA)**: An assembled PCB.
- **Thermal Design Power (TDP)**: The estimated maximum possible/expected power generated in a component by a realistic application. Use Maximum power requirement numbers from Table 15-1.
- **LFM:** Linear Feet per Minute (airflow).

15.6 Thermal Specifications

To ensure proper operation and reliability of the device, the thermal solution must maintain a case temperature at or below the values specified in Table 15-2. System-level or component-level thermal enhancements are required to dissipate the generated heat if the case temperature exceeds the maximum temperatures listed in Table 15-2.

Analysis indicates that real applications are unlikely to cause the device to be at Tcase-max for sustained periods of time. Given that Tcase should reasonably be expected to be a distribution of temperatures, sustained operation at Tcase-max may be indicative that the given thermal solution will also result in situations where Tcase exceeds the specified maximum value. Such thermal designs may affect long-term reliability of the device and the system, and sustained performance at Tcase-max should be evaluated during the thermal design process and steps taken to further reduce the Tcase temperature.

Good system airflow is critical to dissipate the highest possible thermal power. The size and number of fans, vents, and/or ducts, and, their placement in relation to components and airflow channels within the system determine airflow. Acoustic noise constraints may limit the size and types of fans, vents and ducts that can be used in a particular design.

To develop a reliable, cost-effective thermal solution, all of the system variables must be considered. Use system-level thermal characteristics and simulations to account for individual component thermal requirements.

Table 15-2. 82576 Thermal Absolute Maximum Rating

Parameter	Maximum
Tcase-hs[1]	113°C
Tcase-no hs[2]	111°C

[1] Tcase-hs is defined as the maximum case temperature with the default enhanced thermal solution attached.
[2] Tcase-no hs is defined as the maximum case temperature without any thermal enhancement to the package.

15.6.1 Case Temperature

The device is designed to operate properly as long as the Tcase is not exceeded. See Section 15.8.1 for guidelines.

15.7 Thermal Attributes

15.7.1 Designing for Thermal Performance

The appendices of this document give the PCB and system design recommendations required to achieve the thermal performance documented herein.

15.7.2 Typical System Definitions

The following system example is used to generate thermal characteristics data:

- Heat sink case assumes the default enhanced thermal solution. See Section 15.7.6.
- Evaluation board is a standard multi-layer 4s4p 1.0-oz Cu signal layer, 1.0-oz power/ground layer PCB.
- Data at 50LFM and 150LFM is validated against physical samples.

Your design may be different. A larger board size with more than six Cu layers may increase thermal performance.

15.7.3 Package Thermal Characteristics

Figure 15-1 shows the required local ambient temperature versus airflow for a typical system.

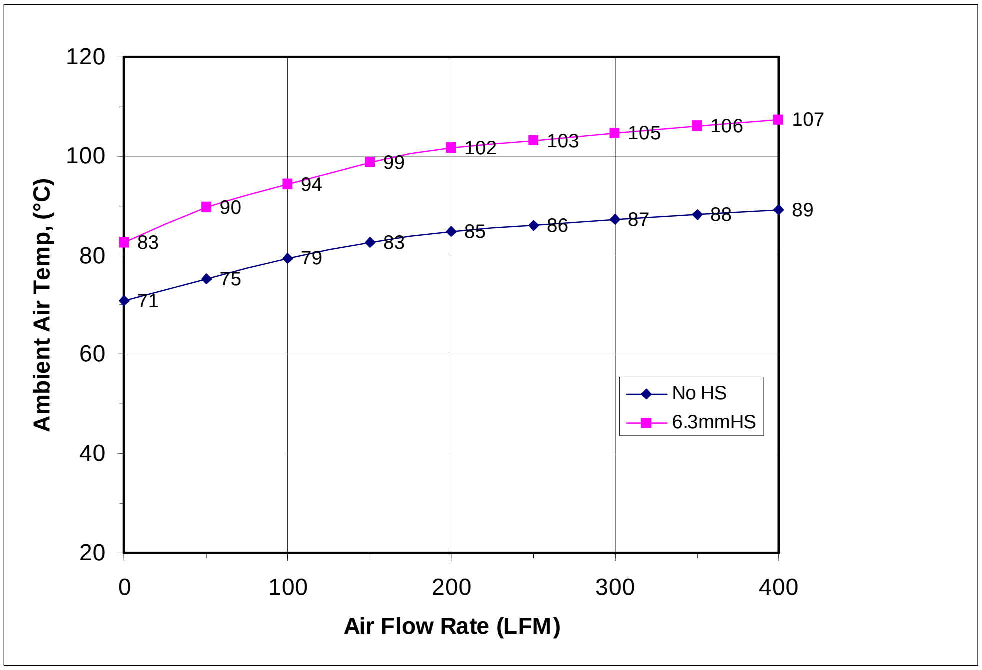


Figure 15-1. Maximum Allowable Ambient Temperature vs. Airflow

Thermal models are available upon request (Flotherm: 2-Resistor, Delphi, or Detailed and Icepak: Detailed). Contact Intel sales.

Table 15-3 shows Tcase as a function of airflow and ambient temperature at the TDP for a typical system and aids in determining the optimum airflow and heat sink combination for the device.

Table 15-3. Expected Tcase (°C) for Heat Sink Attached to TDP

6.3mm Heat Sink									
Tcase max =	113								
Heat Sink Attached							Tcase Max =		113C
85 C amb.	116	109	104	100	97	95	94	92	91
80 C amb.	111	104	99	95	92	90	89	87	86
75 C amb.	106	99	94	90	87	85	84	82	81
70 C amb.	101	94	89	85	82	80	79	77	76
65 C amb.	96	89	84	80	77	75	74	72	71
60 C amb.	91	84	79	75	72	70	69	67	66
55 C amb.	86	79	74	70	67	65	64	62	61
50 C amb.	81	74	69	65	62	60	59	57	56
45 C amb.	76	69	64	60	57	55	54	52	51
Air Flow LFM	0	50	100	150	200	250	300	350	400

The following table shows Tcase as a function of airflow and ambient temperature at the TDP for a typical system and aids in determining the optimum airflow for the device.

Table 15-4. Expected Tcase (°C) For No Heat Sink Attached At TDP

No Heat Sink Attached							Tcase Max =		111C
85 C amb.	124	120	116	113	110	109	108	107	106
80 C amb.	119	115	111	108	105	104	103	102	101
75 C amb.	114	110	106	103	100	99	98	97	96
70 C amb.	109	105	101	98	95	94	93	92	91
65 C amb.	104	100	96	93	90	89	88	87	86
60 C amb.	99	95	91	88	85	84	83	82	81
55 C amb.	94	90	86	83	80	79	78	77	76
50 C amb.	89	85	81	78	75	74	73	72	71
45 C amb.	84	80	76	73	70	69	68	67	66
Air Flow LFM	0	50	100	150	200	250	300	350	400

Note: The underlined values indicate airflow/local ambient combinations that exceed the allowable case temperature for the typical system.

Thermal enhancements (if required) are a method frequently used to improve thermal performance by increasing the component's surface area by attaching a metallic heat sink to the component top. Increasing the surface area of the heat sink reduces the thermal resistance from the heat sink to the air increasing heat transfer.

15.7.4 Clearance

To be effective, a heat sink requires a pocket of air around it free of obstructions. Though each design may have unique mechanical restrictions, recommended clearance zones for a heat sink used with the 82576EA/EB/ES are in Figure 15-2 and Figure 15-3.

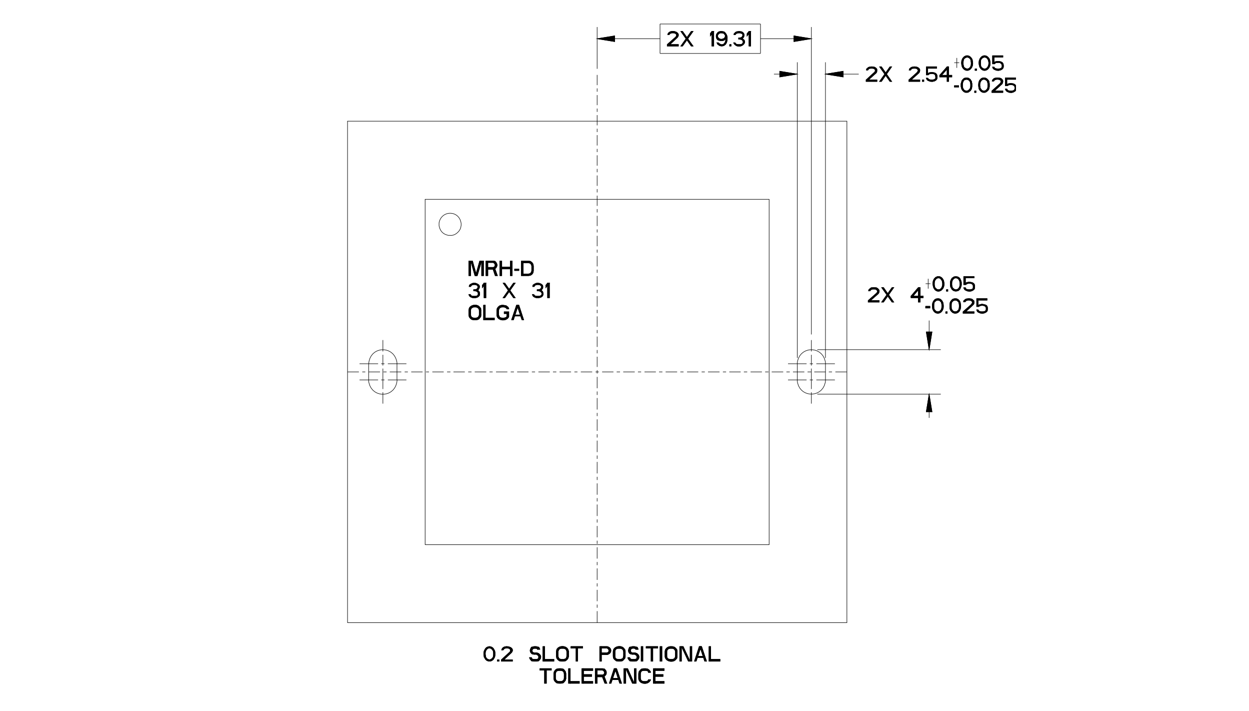


Figure 15-2. 82576EA/EB/ES Heat Sink Volume Restrictions: Primary Side

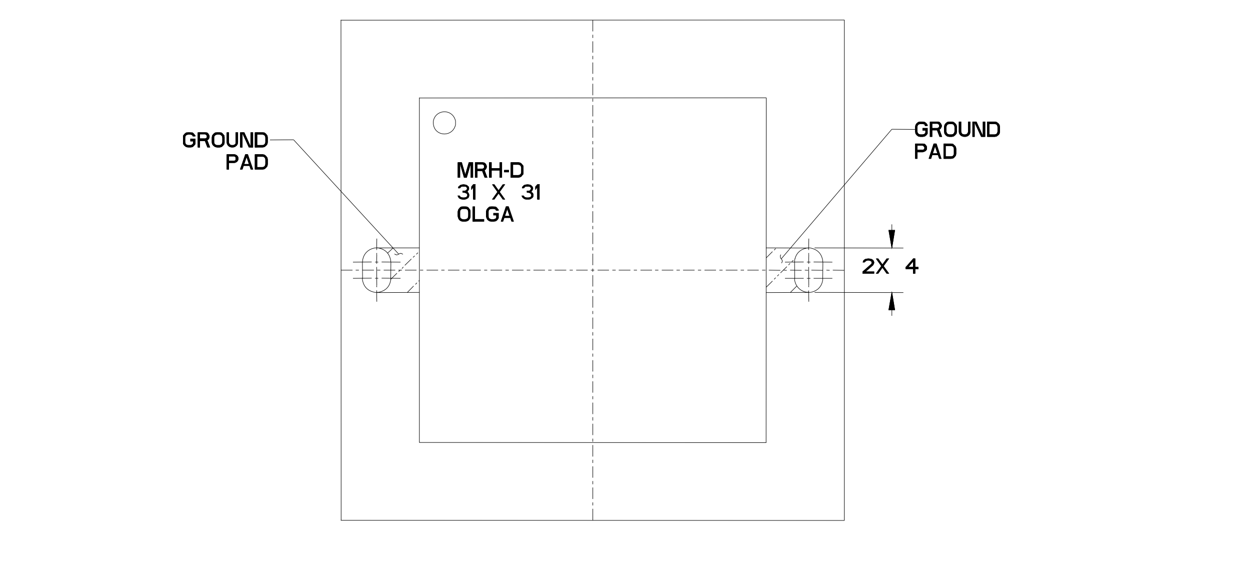


Figure 15-3. 82576EA/EB/ES Heat Sink Volume Restrictions: Secondary Side

15.7.5 Default Enhanced Thermal Solution

If you have no control over the end-user's thermal environment or if you wish to bypass the thermal modeling and evaluation process, use the Default Enhanced Thermal Solution (discussed in the following section). If the case temperature continues to exceed the appropriate value listed in Table 15-2 after implementing the Default Enhanced Thermal Solution, additional cooling is needed, see Figure 15-1. The thermal performance gain may be achieved by improving airflow to the component and/or adding additional thermal enhancements.

15.7.6 Extruded Heat sinks

If required, the following extruded heat sink is the suggested thermal solution. Figure 15-4 shows the heat sink drawing. For equivalent heat sinks and sources, see Section 15.9.

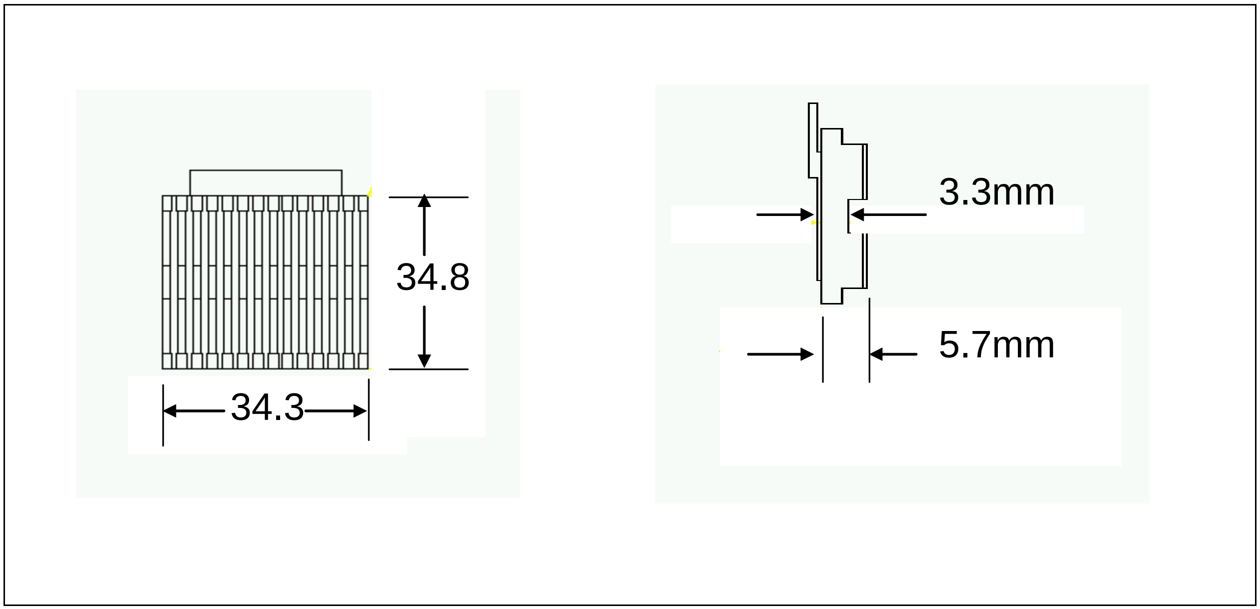


Figure 15-4. 82576 Extruded Heat sink (in millimeters)

15.7.7 Attaching the Extruded Heat sink

The extruded heat sink may be attached using clips with a phase change thermal interface material.

15.7.7.1 Clips

A well-designed clip, in conjunction with a thermal interface material (tape, grease, etc.) often offers the best combination of mechanical stability and rework-ability. Use of a clip requires significant advance planning as mounting holes are required in the PCB. Use non-plated mounting with a grounded annular ring on the solder side of the board surrounding the hole. For a typical low-cost clip, set the annular ring inner diameter to 150 mils and an outer diameter to 300 mils. Define the ring to have at least eight ground connections. Set the solder mask opening for these holes with a radius of 300 mils.

15.7.7.2 Thermal Interface Material (PCM45F)

The recommended thermal interface is PCM45F from Honeywell. The PCM45F thermal interface pads are phase change materials formulated for use in high performance devices requiring minimum thermal resistance for maximum heat sink performance and component reliability. These pads consist of an electrically non-conductive, dry film that softens at device operating temperatures resulting in "grease-like" performance. Alternate recommended TIM is PCM45F from Honeywell for cost saving purposes. However, Intel has not fully validated the PCM45F TIM.

Following the manufacturers recommended attach procedure list for the recommended thermal interface.

1. Ensure that the component surface and heat sink are free from contamination. Using proper safety precaution, clean the package top with a lint-free wipe and Isopropyl Alcohol.
2. Pre heat the heat sink to 50 C. Remove the Honeywell PCM45F from the carrier. For best result, Peel the TIM off of the carrier by peeling back the carrier at 180 degrees.
3. Carefully align the pad, and place it on the heat sink.
4. Apply 10 PSI pressure to the PCM45F pad and let the heat sink cool to room temperature (25C).
5. Remove top liner. Peel back at 180 degrees to prevent voids and achieve best results.
6. Dents and minor scratches in the material will not affect performance since the material is designed to flow at typical operating temperatures. Honeywell pads can be removed for rework using a single-edged razor and then cleaning the surface with isopropyl (IPA) solvent.

Each PCA, system and heat sink combination varies in attach strength. Carefully evaluate the reliability of tape attaches prior to high-volume use.

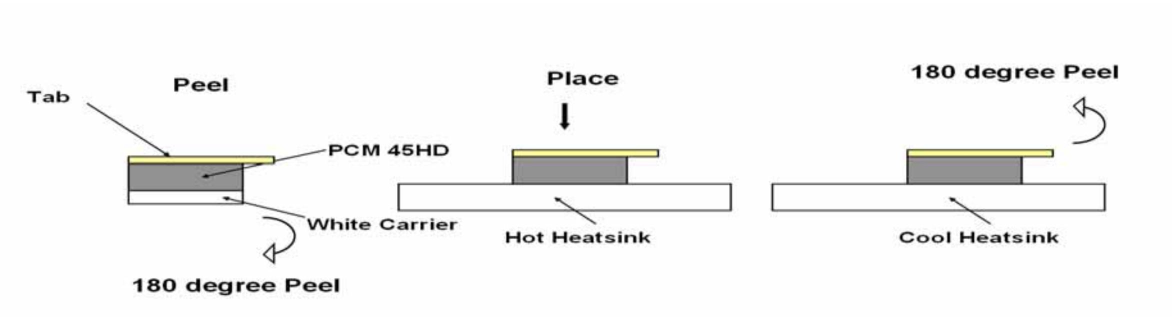


Figure 15-5. PCM45F Attach Process (in roll form)

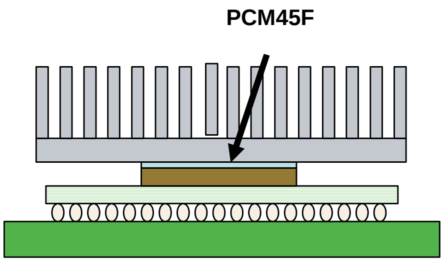


Figure 15-6. Completing the Attach Process

15.7.8 Reliability

Each PCA, system and heat sink combination varies in attach strength and long-term adhesive performance. Carefully evaluate the reliability of the completed assembly prior to high-volume use. Some reliability recommendations are shown in Table 15-5.

Table 15-5. Reliability Validation

Test [1]	Requirement	Pass/Fail Criteria [2]
Mechanical Shock	50G, board level 11 ms trapezoidal pulse, 3 shocks/axis	Visual & Electrical Check
Random Vibration	7.3G, board level 45 minutes/axis, 50 to 2000 Hz	Visual & Electrical Check
High-Temperature Life	85 ^{o}C 2000 hours total Checkpoints occur at 168, 500, 1000, and 2000 hours	Visual & Mechanical Check
Thermal Cycling	Per-Target Environment (for example: -40 ^{o}C to +85 ^{o}C) 500 Cycles	Visual & Mechanical Check
Humidity	85% relative humidity 85 ^{o}C, 1000 hours	Visual & Mechanical Check

1. Perform the above tests on a sample size of at least 12 assemblies from 3 lots of material (total = 36 assemblies).
2. Additional Pass/Fail Criteria can be added at your discretion.

15.7.9 Thermal Interface Management for Heat-Sink Solutions

To optimize the 82576 heat sink design, it is important to understand the interface between the exposed die and the heat sink base. Specifically, thermal conductivity effectiveness depends on the following:

- Bond line thickness
- Interface material area
- Interface material thermal conductivity

15.7.9.1 Bond Line Management

The gap between the exposed die and the heat sink base impacts the heat-sink solution performance. The larger the gap between the two surfaces, the greater the thermal resistance. The thickness of the gap is determined by the flatness of both the heat sink base and the exposed die, plus the thickness of the thermal interface material (for example, PSA, thermal grease, epoxy) used to join the two surfaces.

Planarity of the 82576 package is 8 mils (in accordance with JEDEC specifications).

15.7.9.2 Interface Material Performance

The following two factors impact the performance of the interface material between the exposed die and the heat sink base:

- Thermal resistance of the material
- Wetting/filling characteristics of the material

15.7.9.2.1 Thermal Resistance of Material

Thermal resistance describes the ability of the thermal interface material to transfer heat from one surface to another. The higher the thermal resistance, the less efficient is the heat transfer. The thermal resistance of the interface material has a significant impact on the thermal performance of the overall thermal solution. With a higher thermal resistance, there will be a larger temperature drop across the interface.

15.7.9.2.2 Wetting/Filling Characteristics of Material

The wetting/filling characteristic of the thermal interface material is its ability to fill the gap between the exposed die top surface and the heat sink. Since air is an extremely poor thermal conductor, the more completely the interface material fills the gaps, the lower the temperature-drop across the interface, increasing the efficiency of the thermal solution.

15.8 Measurements for Thermal Specifications

Determining the thermal properties of the system requires careful case temperature measurements. Guidelines for measuring the 82576 case temperature are provided in Section 15.8.1.

15.8.1 Case Temperature Measurements

Maintain the 82576 Tcase at or below the maximum case temperatures listed in Table 15-2 to ensure functionality and reliability. Special care is required when measuring the case temperature to ensure an accurate temperature measurement. Use the following guidelines when making case measurements:

- Measure the surface temperature of the case in the geometric center of the case top.
- Calibrate the thermocouples used to measure Tcase before making temperature measurements.
- Use 36-gauge (maximum) K-type thermocouples.

Care must be taken to avoid introducing errors into the measurements when measuring a surface temperature that is a different temperature from the surrounding local ambient air. Measurement errors may be due to a poor thermal contact between the thermocouple junction and the surface of the package, heat loss by radiation, convection, conduction through thermocouple leads, and/or contact between the thermocouple cement and the heat-sink base (if used).

15.8.1.1 Attaching the Thermocouple (No Heat Sink)

The following approach is recommended to minimize measurement errors for attaching the thermocouple with no heat sink:

- Use 36 gauge or smaller diameter K type thermocouples.
- Ensure that the thermocouple has been properly calibrated.
- Attach the thermocouple bead or junction to the top surface of the package (case) in the center of the silicon die using high thermal conductivity cement.
- **It is critical that the entire thermocouple lead be butted tightly to the exposed die.**
- Attach the thermocouple at a 0° angle if there is no interference with the thermocouple attach location or leads (Figure 15-7). This is the preferred method and is recommended for use with non-enhanced packages.

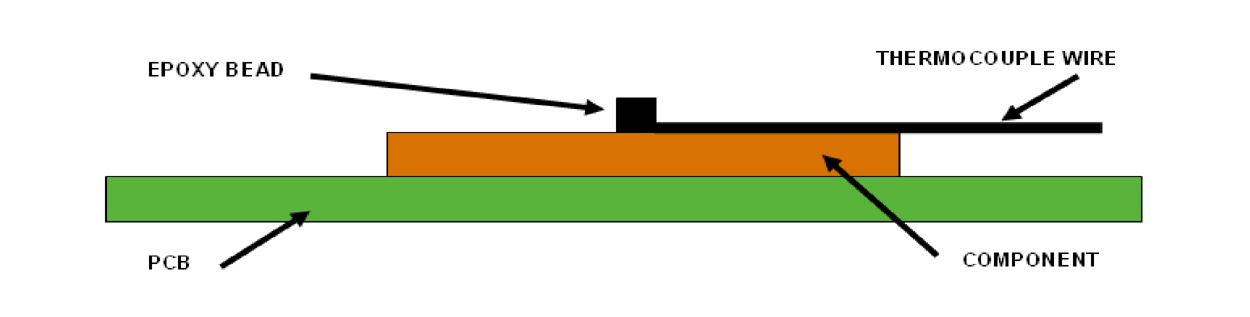


Figure 15-7. Technique for Measuring Tcase with 0° Angle Attachment, No Heat Sink

15.8.1.2 Attaching the Thermocouple (Heat Sink)

The following approach is recommended to minimize measurement errors for attaching the thermocouple with heat sink:

- Use 36 gauge or smaller diameter K-type thermocouples.
- Ensure that the thermocouple is properly calibrated.
- Attach the thermocouple bead or junction to the case's top surface in the geometric center using high thermal conductivity cement.
- **It is critical that the entire thermocouple lead be butted tightly against the case.**
- Attach the thermocouple at a 90° angle if there is no interference with the thermocouple attach location or leads (refer to Figure 15-7). This is the preferred method and is recommended for use with packages with heat sinks.
- For testing purposes, a hole (no larger than 0.150" in diameter) must be drilled vertically through the center of the heat sink to route the thermocouple wires out.
- Ensure there is no contact between the thermocouple cement and heat sink base. Any contact affects the thermocouple reading.

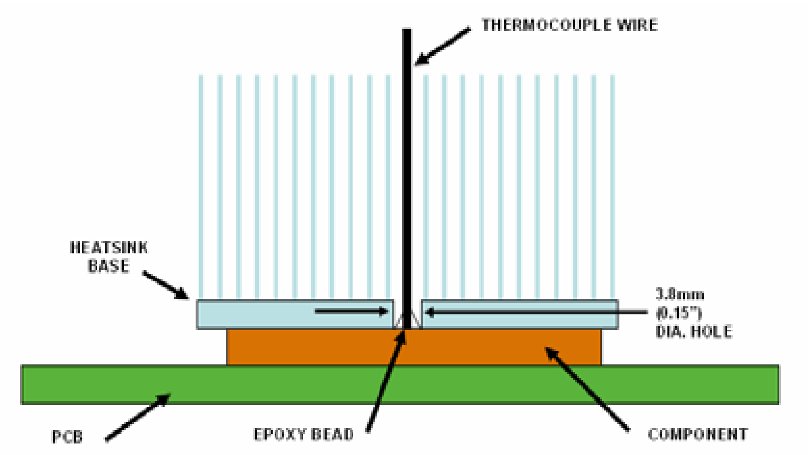


Figure 15-8. Technique for Measuring Tcase with 90° Angle Attachment

15.9 Heat Sink and Attach Suppliers

Table 15-6. Hint Sink and Attach Suppliers

Part	Part number	Supplier	Contact information
Heatsink	728443-001	Foxconn	Hon Hai Precision Industry Co Ltd Contact: Susiey Chen susiey.chen@foxconn.com
Retention Mechanism	C63585-0C1	CCI	Chaun-Choung Tech. Corp Contact: Monica Chih 12F NO123-1 Hsing-De Rd. Sanchung, Taipei, Taiwan Tel: 886-2-29952666 Fax: 886-2-29958258 monica_chih@ccic.com.tw
Thermal interface	PCM45F included with heatsink size = 20mm^2	Honeywell	North America Technical Contact: Paula Knoll 1349 Moffett Park Dr. Sunnyvale, CA 94089 Cell: 1-858-705-1274 Business: 858-279-2956 paula.knoll@honeywell.com

15.10 PCB Guidelines

The following general PCB design guidelines are recommended to maximize the thermal performance of FCBGA packages:

- When connecting ground (thermal) vias to the ground planes, do not use thermal-relief patterns.
- Thermal-relief patterns are designed to limit heat transfer between the vias and the copper planes, thus constricting the heat flow path from the component to the ground planes in the PCB.
- As board temperature also has an effect on the thermal performance of the package, avoid placing the 82576 adjacent to high power dissipation devices.
- If airflow exists, locate the components in the mainstream of the airflow path for maximum thermal performance. Avoid placing the components downstream, behind larger devices or devices with heat sinks that obstruct the air flow or supply excessively heated air.

The above guidelines are not all inclusive and are defined to give you known, good design practices to maximize the thermal performance of the components.

§ §

16. Schematics

The following pages provide reference design schematics.

82576
REFERENCE DESIGN
(SERDES/FIBER/SFP)

REVISION HISTORY

0.1 ORIGINAL VERSION

1.0 UPDATED BASED ON THE LATEST PRODUCT REQUIREMENTS.

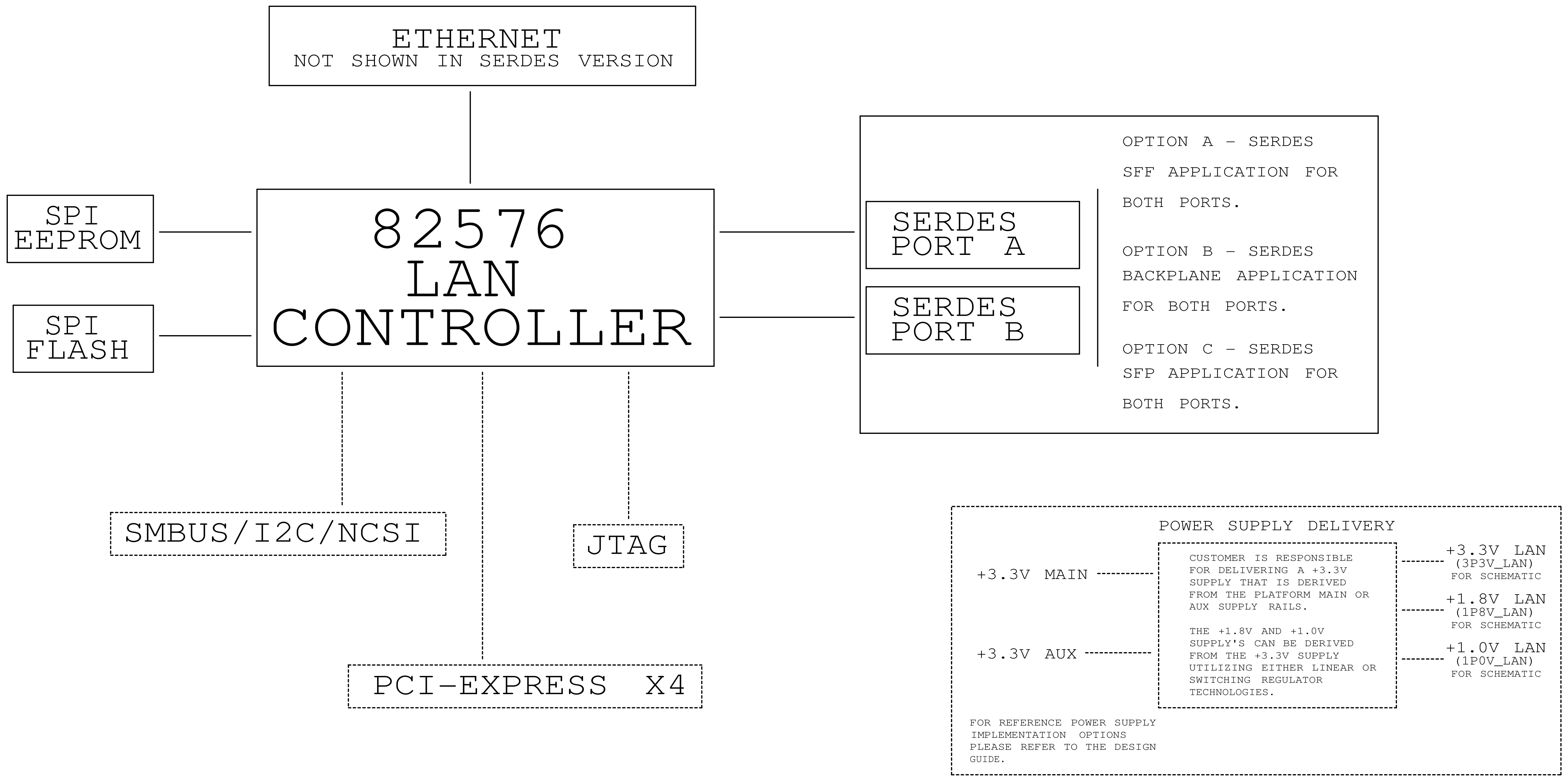
2.0 FINAL

PAGE INDEX

1 – TITLE PAGE

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000–000	REV 2.0	DATE 04/06/2008	SHEET 1

FUNCTIONAL BLOCK DIAGRAM


ETHERNET
NOT SHOWN IN SERDES VERSION
SPI EEPROM
SPI FLASH
82576 LAN CONTROLLER
SERDES PORT A
SERDES PORT B
OPTION A - SERDES SFF APPLICATION FOR BOTH PORTS.
OPTION B - SERDES BACKPLANE APPLICATION FOR BOTH PORTS.
OPTION C - SERDES SFP APPLICATION FOR BOTH PORTS.
SMBUS/I2C/NCSI
JTAG
PCI-EXPRESS X4
POWER SUPPLY DELIVERY
+3.3V MAIN
+3.3V AUX
CUSTOMER IS RESPONSIBLE FOR DELIVERING A +3.3V SUPPLY THAT IS DERIVED FROM THE PLATFORM MAIN OR AUX SUPPLY RAILS.
THE +1.8V AND +1.0V SUPPLY'S CAN BE DERIVED FROM THE +3.3V SUPPLY UTILIZING EITHER LINEAR OR SWITCHING REGULATOR TECHNOLOGIES.
+3.3V LAN
(3P3V_LAN)
FOR SCHEMATIC
+1.8V LAN
(1P8V_LAN)
FOR SCHEMATIC
+1.0V LAN
(1P0V_LAN)
FOR SCHEMATIC
FOR REFERENCE POWER SUPPLY IMPLEMENTATION OPTIONS PLEASE REFER TO THE DESIGN GUIDE.

LEGEND

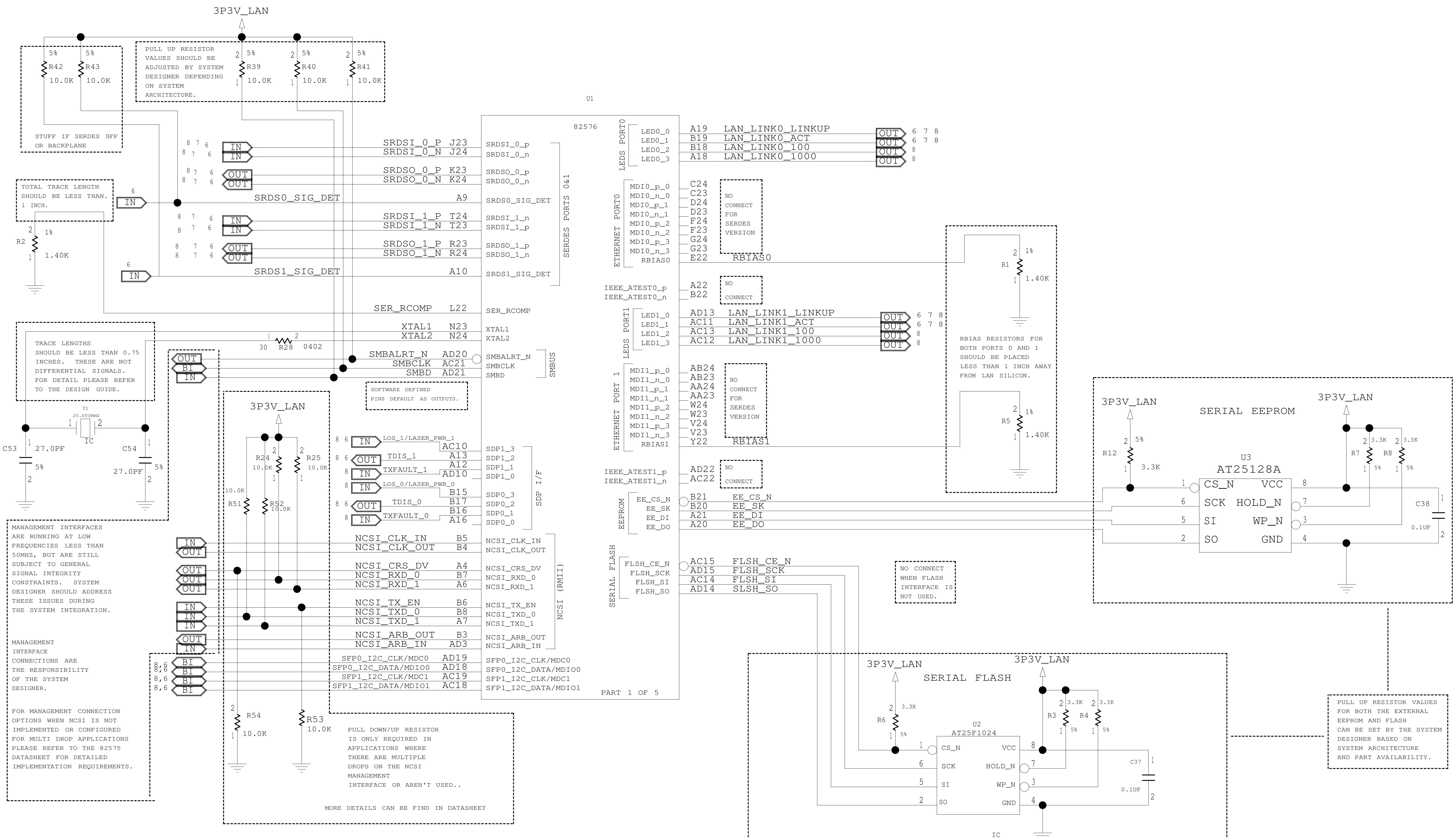
— SOLID LINE INDICATES THAT THE CONNECTIONS ARE SHOWN IN REFERENCE SCHEMATIC.

---- DASHED LINE INDICATES THAT THE CONNECTIONS ARE NOT SHOWN IN REFERENCE SCHEMATIC. THESE CONNECTIONS ARE OWNED BY THE SYSTEM DESIGNER.

2 - FUNCTIONAL BLOCK DIAGRAM

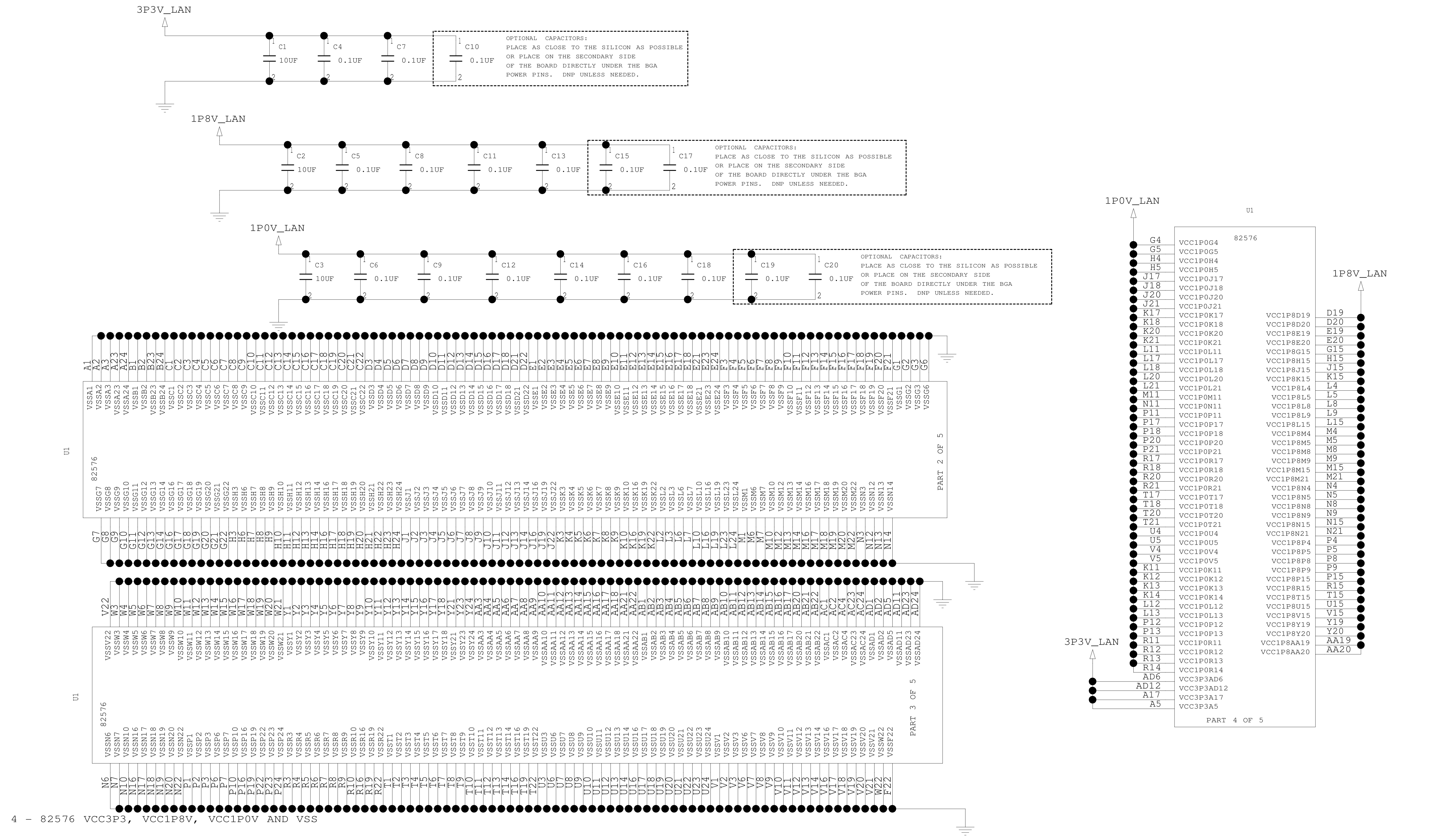
LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/06/2008	2


3P3V_LAN
R42 10.0K 5%
R43 10.0K 5%
STUFF IF SERDES SFP OR BACKPLANE
PULL UP RESISTOR VALUES SHOULD BE ADJUSTED BY SYSTEM DESIGNER DEPENDING ON SYSTEM ARCHITECTURE.
R39 10.0K 5%
R40 10.0K 5%
R41 10.0K 5%
U1
82576
TOTAL TRACE LENGTH SHOULD BE LESS THAN. 1 INCH.
R2 1.40K 1%
IN
OUT
BI
SRDSI_0_P J23 SRDSI_0_p
SRDSI_0_N J24 SRDSI_0_n
SRDSO_0_P K23 SRDSO_0_p
SRDSO_0_N K24 SRDSO_0_n
SRDS0_SIG_DET A9 SRDS0_SIG_DET
SRDSI_1_P T24 SRDSI_1_n
SRDSI_1_N T23 SRDSI_1_p
SRDSO_1_P R23 SRDSO_1_p
SRDSO_1_N R24 SRDSO_1_n
SRDS1_SIG_DET A10 SRDS1_SIG_DET
SERDES PORTS 0&1
SER_RCOMP L22 SER_RCOMP
XTAL1 N23 XTAL1
XTAL2 N24 XTAL2
R28 0402 30
TRACE LENGTHS SHOULD BE LESS THAN 0.75 INCHES. THESE ARE NOT DIFFERENTIAL SIGNALS. FOR DETAIL PLEASE REFER TO THE DESIGN GUIDE.
Y1 25.000MHZ
IC
C53 27.0PF 5%
C54 27.0PF 5%
SMBALRT_N AD20 SMBALRT_N
SMBCLK AC21 SMBCLK
SMBD AD21 SMBD
SMBUS
SOFTWARE DEFINED PINS DEFAULT AS OUTPUTS.
LOS_1/LASER_PWR_1 AC10 SDP1_3
TDIS_1 A13 SDP1_2
TXFAULT_1 A12 SDP1_1
AD10 SDP1_0
LOS_0/LASER_PWR_0 B15 SDP0_3
TDIS_0 B17 SDP0_2
B16 SDP0_1
TXFAULT_0 A16 SDP0_0
SDP I/F
R24 10.0K
R25 10.0K
R51 10.0K
R52 10.0K
NCSI_CLK_IN B5 NCSI_CLK_IN
NCSI_CLK_OUT B4 NCSI_CLK_OUT
NCSI_CRS_DV A4 NCSI_CRS_DV
NCSI_RXD_0 B7 NCSI_RXD_0
NCSI_RXD_1 A6 NCSI_RXD_1
NCSI_TX_EN B6 NCSI_TX_EN
NCSI_TXD_0 B8 NCSI_TXD_0
NCSI_TXD_1 A7 NCSI_TXD_1
NCSI_ARB_OUT B3 NCSI_ARB_OUT
NCSI_ARB_IN AD3 NCSI_ARB_IN
NCSI (RMII)
SFP0_I2C_CLK/MDC0 AD19 SFP0_I2C_CLK/MDC0
SFP0_I2C_DATA/MDIO0 AD18 SFP0_I2C_DATA/MDIO0
SFP1_I2C_CLK/MDC1 AC19 SFP1_I2C_CLK/MDC1
SFP1_I2C_DATA/MDIO1 AC18 SFP1_I2C_DATA/MDIO1
PART 1 OF 5
MANAGEMENT INTERFACES ARE RUNNING AT LOW FREQUENCIES LESS THAN 50MHZ, BUT ARE STILL SUBJECT TO GENERAL SIGNAL INTEGRITY CONSTRAINTS. SYSTEM DESIGNER SHOULD ADDRESS THESE ISSUES DURING THE SYSTEM INTEGRATION.
MANAGEMENT INTERFACE CONNECTIONS ARE THE RESPONSIBILITY OF THE SYSTEM DESIGNER.
FOR MANAGEMENT CONNECTION OPTIONS WHEN NCSI IS NOT IMPLEMENTED OR CONFIGURED FOR MULTI DROP APPLICATIONS PLEASE REFER TO THE 82575 DATASHEET FOR DETAILED IMPLEMENTATION REQUIREMENTS.
R54 10.0K
R53 10.0K
PULL DOWN/UP RESISTOR IS ONLY REQUIRED IN APPLICATIONS WHERE THERE ARE MULTIPLE DROPS ON THE NCSI MANAGEMENT INTERFACE OR AREN'T USED..
MORE DETAILS CAN BE FIND IN DATASHEET
LEDS PORT0
LED0_0 A19 LAN_LINK0_LINKUP
LED0_1 B19 LAN_LINK0_ACT
LED0_2 B18 LAN_LINK0_100
LED0_3 A18 LAN_LINK0_1000
ETHERNET PORT0
MDI0_p_0 C24
MDI0_n_0 C23
MDI0_p_1 D24
MDI0_n_1 D23
MDI0_p_2 F24
MDI0_n_2 F23
MDI0_p_3 G24
MDI0_n_3 G23
RBIAS0 E22 RBIAS0
NO CONNECT FOR SERDES VERSION
IEEE_ATEST0_p A22
IEEE_ATEST0_n B22
NO CONNECT
LEDS PORT1
LED1_0 AD13 LAN_LINK1_LINKUP
LED1_1 AC11 LAN_LINK1_ACT
LED1_2 AC13 LAN_LINK1_100
LED1_3 AC12 LAN_LINK1_1000
ETHERNET PORT 1
MDI1_p_0 AB24
MDI1_n_0 AB23
MDI1_p_1 AA24
MDI1_n_1 AA23
MDI1_p_2 W24
MDI1_n_2 W23
MDI1_p_3 V24
MDI1_n_3 V23
RBIAS1 Y22 RBIAS1
IEEE_ATEST1_p AD22
IEEE_ATEST1_n AC22
EEPROM
EE_CS_N B21 EE_CS_N
EE_SK B20 EE_SK
EE_DI A21 EE_DI
EE_DO A20 EE_DO
SERIAL FLASH
FLSH_CE_N AC15 FLSH_CE_N
FLSH_SCK AD15 FLSH_SCK
FLSH_SI AC14 FLSH_SI
FLSH_SO AD14 SLSH_SO
NO CONNECT WHEN FLASH INTERFACE IS NOT USED.
R1 1.40K 1%
R5 1.40K 1%
RBIAS RESISTORS FOR BOTH PORTS 0 AND 1 SHOULD BE PLACED LESS THAN 1 INCH AWAY FROM LAN SILICON.
SERIAL EEPROM
R12 3.3K 5%
U3
AT25128A
CS_N VCC
SCK HOLD_N
SI WP_N
SO GND
R7 3.3K 5%
R8 3.3K 5%
C38 0.1UF
SERIAL FLASH
R6 3.3K 5%
U2
AT25F1024
R3 3.3K 5%
R4 3.3K 5%
C37 0.1UF
IC
PULL UP RESISTOR VALUES FOR BOTH THE EXTERNAL EEPROM AND FLASH CAN BE SET BY THE SYSTEM DESIGNER BASED ON SYSTEM ARCHITECTURE AND PART AVAILABILITY.

3 - 82576 MDI, LEDS, CRYSTAL, SERDES, SFP, NCSI, SMBUS, EEPROM AND FLASH

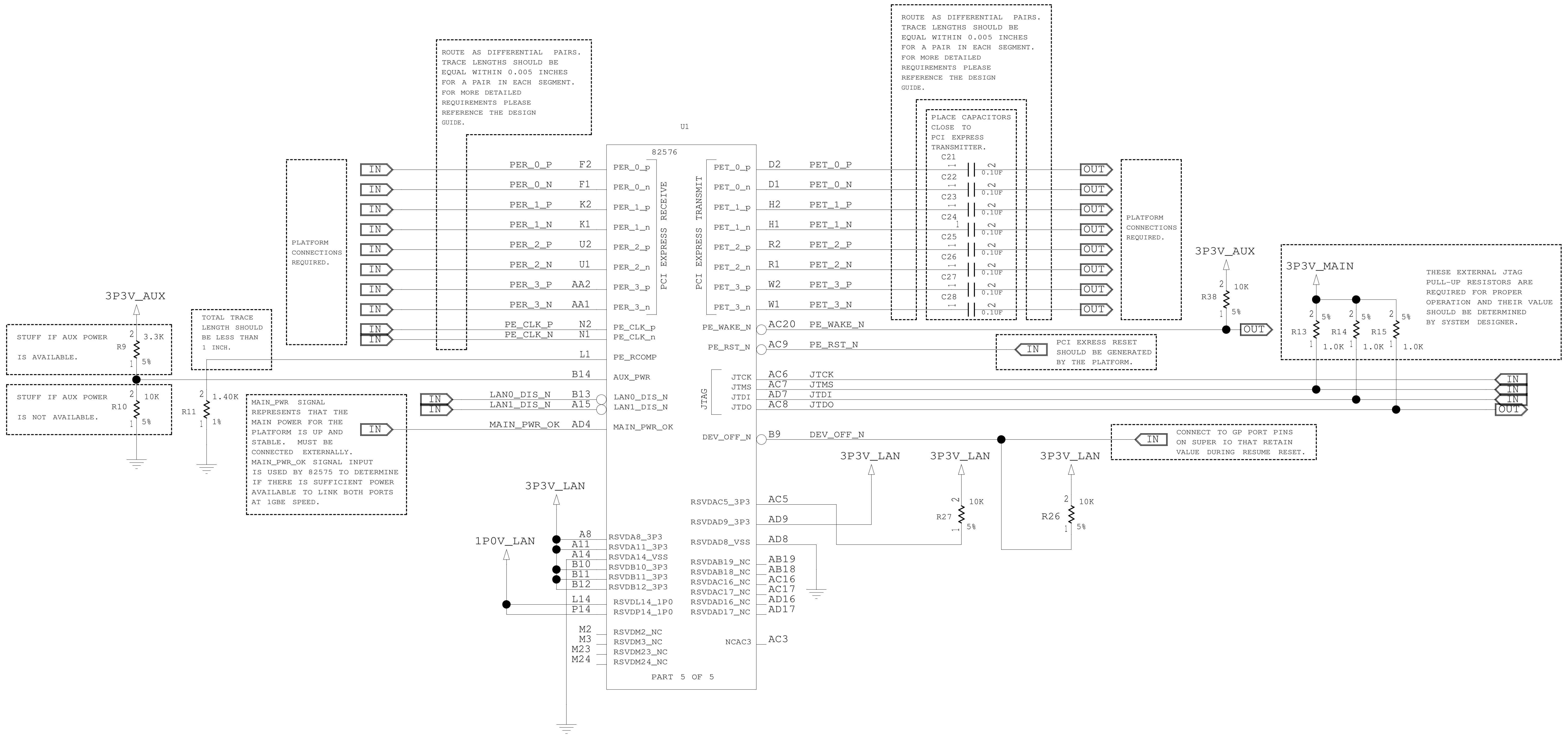
LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000-000	REV 2.0	DATE 04/06/2008	SHEET 3


3P3V_LAN
C1 10UF
C4 0.1UF
C7 0.1UF
C10 0.1UF
OPTIONAL CAPACITORS:
PLACE AS CLOSE TO THE SILICON AS POSSIBLE
OR PLACE ON THE SECONDARY SIDE
OF THE BOARD DIRECTLY UNDER THE BGA
POWER PINS. DNP UNLESS NEEDED.
1P8V_LAN
C2 10UF
C5 0.1UF
C8 0.1UF
C11 0.1UF
C13 0.1UF
C15 0.1UF
C17 0.1UF
OPTIONAL CAPACITORS:
PLACE AS CLOSE TO THE SILICON AS POSSIBLE
OR PLACE ON THE SECONDARY SIDE
OF THE BOARD DIRECTLY UNDER THE BGA
POWER PINS. DNP UNLESS NEEDED.
1P0V_LAN
C3 10UF
C6 0.1UF
C9 0.1UF
C12 0.1UF
C14 0.1UF
C16 0.1UF
C18 0.1UF
C19 0.1UF
C20 0.1UF
OPTIONAL CAPACITORS:
PLACE AS CLOSE TO THE SILICON AS POSSIBLE
OR PLACE ON THE SECONDARY SIDE
OF THE BOARD DIRECTLY UNDER THE BGA
POWER PINS. DNP UNLESS NEEDED.
U1
82576
PART 2 OF 5
U1
82576
PART 3 OF 5
1P0V_LAN
U1
82576
PART 4 OF 5
1P8V_LAN
3P3V_LAN

4 - 82576 VCC3P3, VCC1P8V, VCC1P0V AND VSS

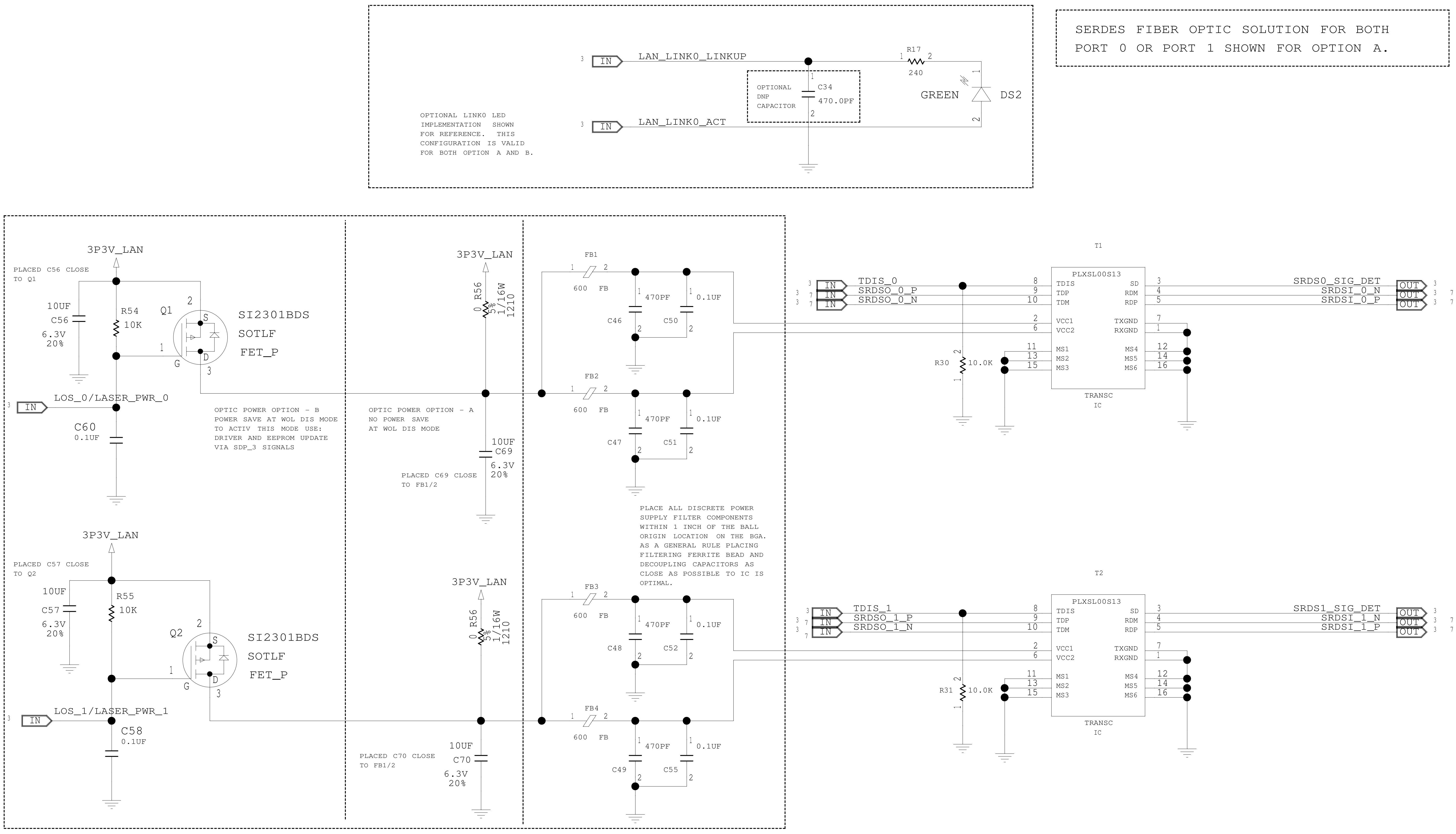
LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/06/2008	4



5 - 82576 PCI EXPRESS, JTAG, AUX POWER, LAN DISABLE, MAIN POWER OK AND DEVICE OFF

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
	82576 REF SCHEMATIC	B		000000-000	2.0	04/06/2008	5



SERDES FIBER OPTIC SOLUTION FOR BOTH PORT 0 OR PORT 1 SHOWN FOR OPTION A.

OPTIONAL LINK0 LED IMPLEMENTATION SHOWN FOR REFERENCE. THIS CONFIGURATION IS VALID FOR BOTH OPTION A AND B.

OPTIONAL DNP CAPACITOR

PLACED C56 CLOSE TO Q1

PLACED C57 CLOSE TO Q2

OPTIC POWER OPTION - B POWER SAVE AT WOL DIS MODE TO ACTIV THIS MODE USE: DRIVER AND EEPROM UPDATE VIA SDP_3 SIGNALS

OPTIC POWER OPTION - A NO POWER SAVE AT WOL DIS MODE

PLACED C69 CLOSE TO FB1/2

PLACED C70 CLOSE TO FB1/2

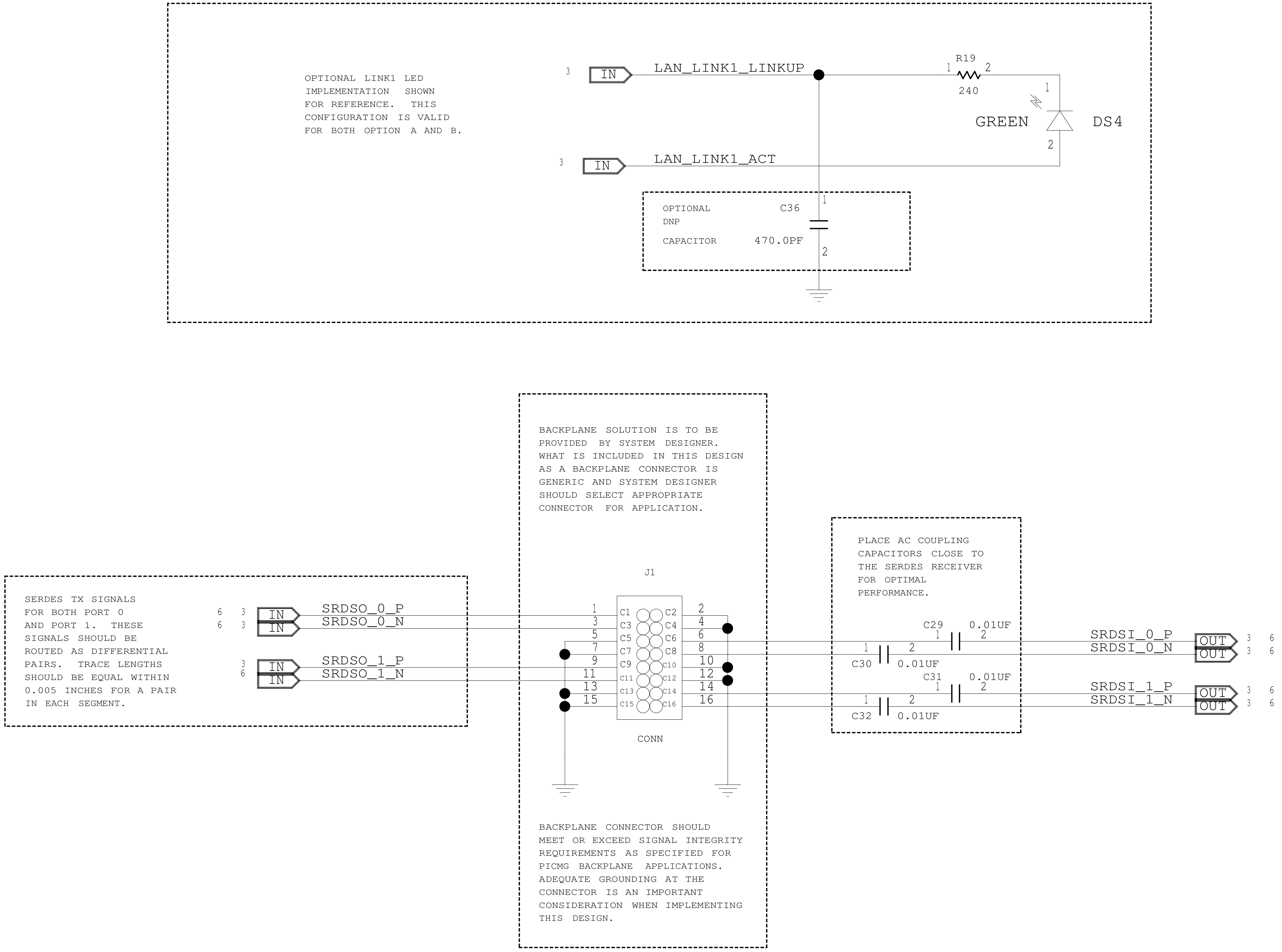
PLACE ALL DISCRETE POWER SUPPLY FILTER COMPONENTS WITHIN 1 INCH OF THE BALL ORIGIN LOCATION ON THE BGA. AS A GENERAL RULE PLACING FILTERING FERRITE BEAD AND DECOUPLING CAPACITORS AS CLOSE AS POSSIBLE TO IC IS OPTIMAL.

6 - SERDES FIBER OPTIC OPTION A - SHOWS IMPLEMENTATION FOR BOTH PORT 0 AND 1.

LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/06/2008	6

SERDES BACKPLANE CONNECTIONS SHOWN FOR BOTH PORT 0 AND PORT 1 FOR OPTION B.



OPTIONAL LINK1 LED IMPLEMENTATION SHOWN FOR REFERENCE. THIS CONFIGURATION IS VALID FOR BOTH OPTION A AND B.

OPTIONAL DNP CAPACITOR

SERDES TX SIGNALS FOR BOTH PORT 0 AND PORT 1. THESE SIGNALS SHOULD BE ROUTED AS DIFFERENTIAL PAIRS. TRACE LENGTHS SHOULD BE EQUAL WITHIN 0.005 INCHES FOR A PAIR IN EACH SEGMENT.

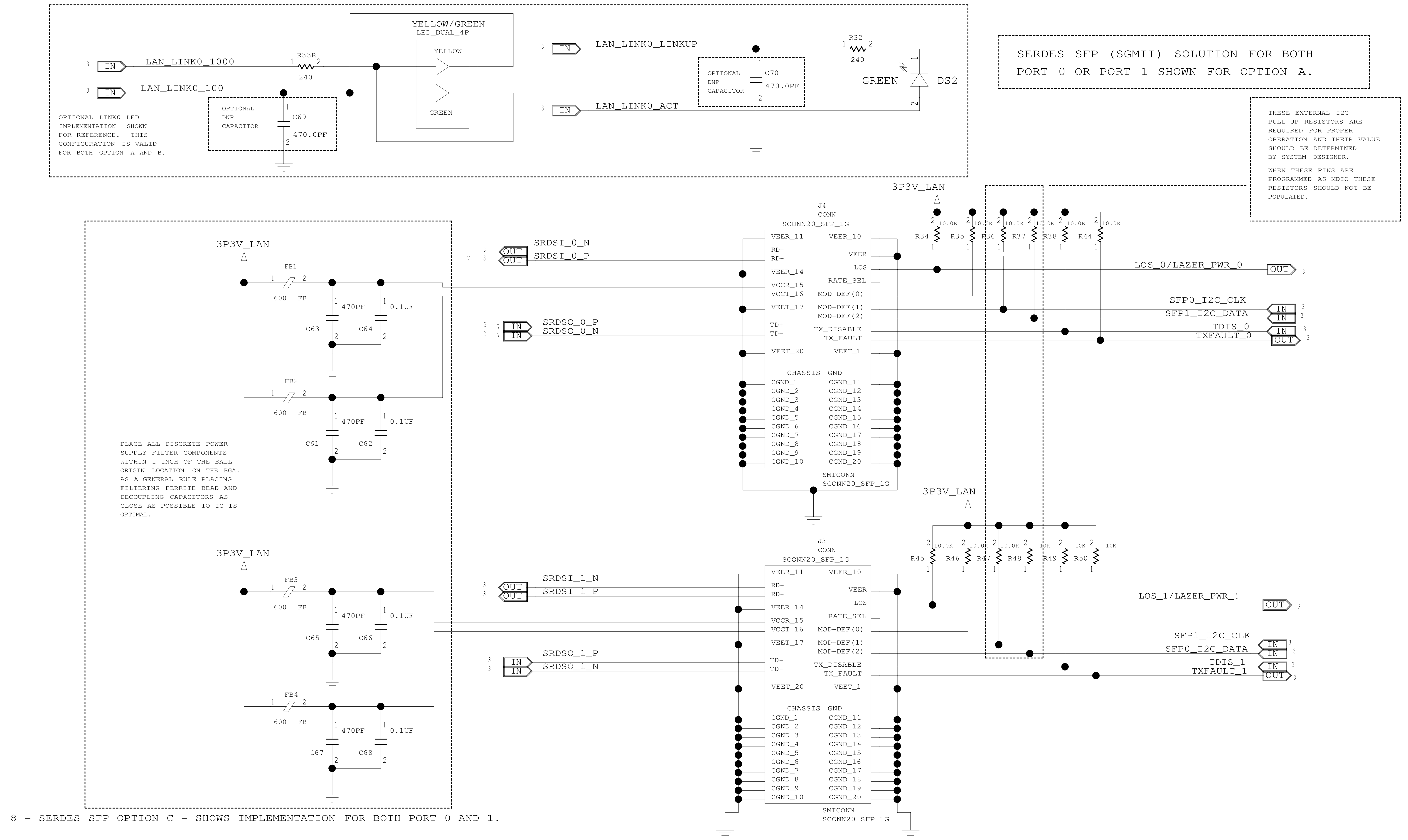
BACKPLANE SOLUTION IS TO BE PROVIDED BY SYSTEM DESIGNER. WHAT IS INCLUDED IN THIS DESIGN AS A BACKPLANE CONNECTOR IS GENERIC AND SYSTEM DESIGNER SHOULD SELECT APPROPRIATE CONNECTOR FOR APPLICATION.

BACKPLANE CONNECTOR SHOULD MEET OR EXCEED SIGNAL INTEGRITY REQUIREMENTS AS SPECIFIED FOR PICMG BACKPLANE APPLICATIONS. ADEQUATE GROUNDING AT THE CONNECTOR IS AN IMPORTANT CONSIDERATION WHEN IMPLEMENTING THIS DESIGN.

PLACE AC COUPLING CAPACITORS CLOSE TO THE SERDES RECEIVER FOR OPTIMAL PERFORMANCE.

7 - SERDES BACKPLANE OPTION B - SHOWS IMPLEMENTATION FOR BOTH PORT 0 AND 1.

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000-000	REV 2.0	DATE 04/06/2008	SHEET 7



OPTIONAL LINK0 LED IMPLEMENTATION SHOWN FOR REFERENCE. THIS CONFIGURATION IS VALID FOR BOTH OPTION A AND B.

SERDES SFP (SGMII) SOLUTION FOR BOTH PORT 0 OR PORT 1 SHOWN FOR OPTION A.

THESE EXTERNAL I2C PULL-UP RESISTORS ARE REQUIRED FOR PROPER OPERATION AND THEIR VALUE SHOULD BE DETERMINED BY SYSTEM DESIGNER.

WHEN THESE PINS ARE PROGRAMMED AS MDIO THESE RESISTORS SHOULD NOT BE POPULATED.

PLACE ALL DISCRETE POWER SUPPLY FILTER COMPONENTS WITHIN 1 INCH OF THE BALL ORIGIN LOCATION ON THE BGA. AS A GENERAL RULE PLACING FILTERING FERRITE BEAD AND DECOUPLING CAPACITORS AS CLOSE AS POSSIBLE TO IC IS OPTIMAL.

8 - SERDES SFP OPTION C - SHOWS IMPLEMENTATION FOR BOTH PORT 0 AND 1.

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000-000	REV 2.0	DATE 04/06/2008	SHEET 8

82576
REFERENCE DESIGN
(COPPER)

PAGE INDEX

REVISION HISTORY

0.1 ORIGINAL VERSION STARTED WITH 82575 AND UPDATED WITH 82576 PART.

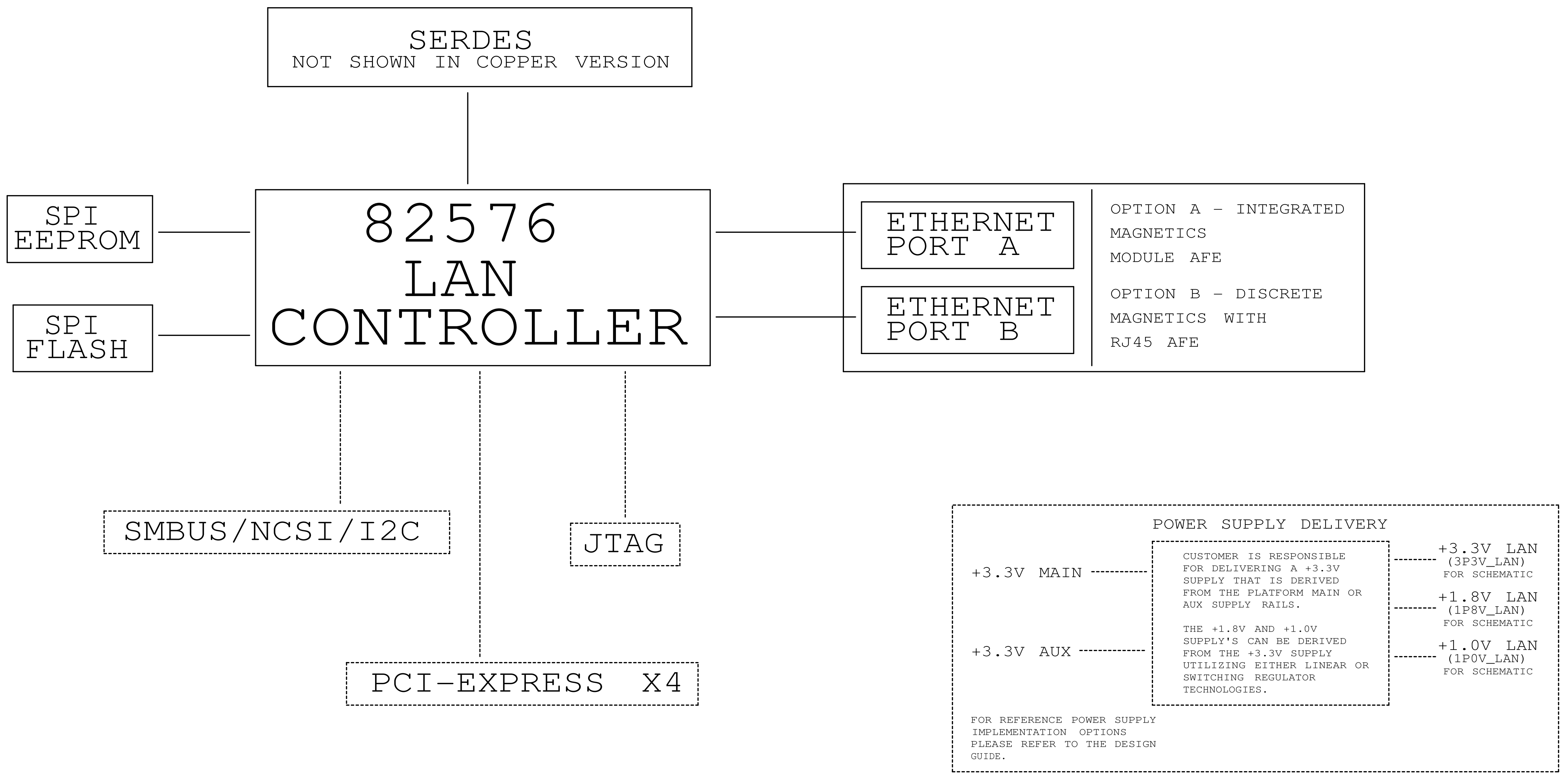
1.0 UPDATED FOR BASED ON LATEST CHANGES.

2.0 FINAL

1 - TITLE PAGE

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000-000	REV 2.0	DATE 04/15/2008	SHEET 1

FUNCTIONAL BLOCK DIAGRAM


SERDES
NOT SHOWN IN COPPER VERSION
SPI EEPROM
SPI FLASH
82576 LAN CONTROLLER
ETHERNET PORT A
ETHERNET PORT B
OPTION A - INTEGRATED MAGNETICS MODULE AFE
OPTION B - DISCRETE MAGNETICS WITH RJ45 AFE
SMBUS/NCSI/I2C
JTAG
PCI-EXPRESS X4
POWER SUPPLY DELIVERY
+3.3V MAIN
+3.3V AUX
CUSTOMER IS RESPONSIBLE FOR DELIVERING A +3.3V SUPPLY THAT IS DERIVED FROM THE PLATFORM MAIN OR AUX SUPPLY RAILS.
THE +1.8V AND +1.0V SUPPLY'S CAN BE DERIVED FROM THE +3.3V SUPPLY UTILIZING EITHER LINEAR OR SWITCHING REGULATOR TECHNOLOGIES.
+3.3V LAN
(3P3V_LAN)
FOR SCHEMATIC
+1.8V LAN
(1P8V_LAN)
FOR SCHEMATIC
+1.0V LAN
(1P0V_LAN)
FOR SCHEMATIC
FOR REFERENCE POWER SUPPLY IMPLEMENTATION OPTIONS PLEASE REFER TO THE DESIGN GUIDE.

LEGEND

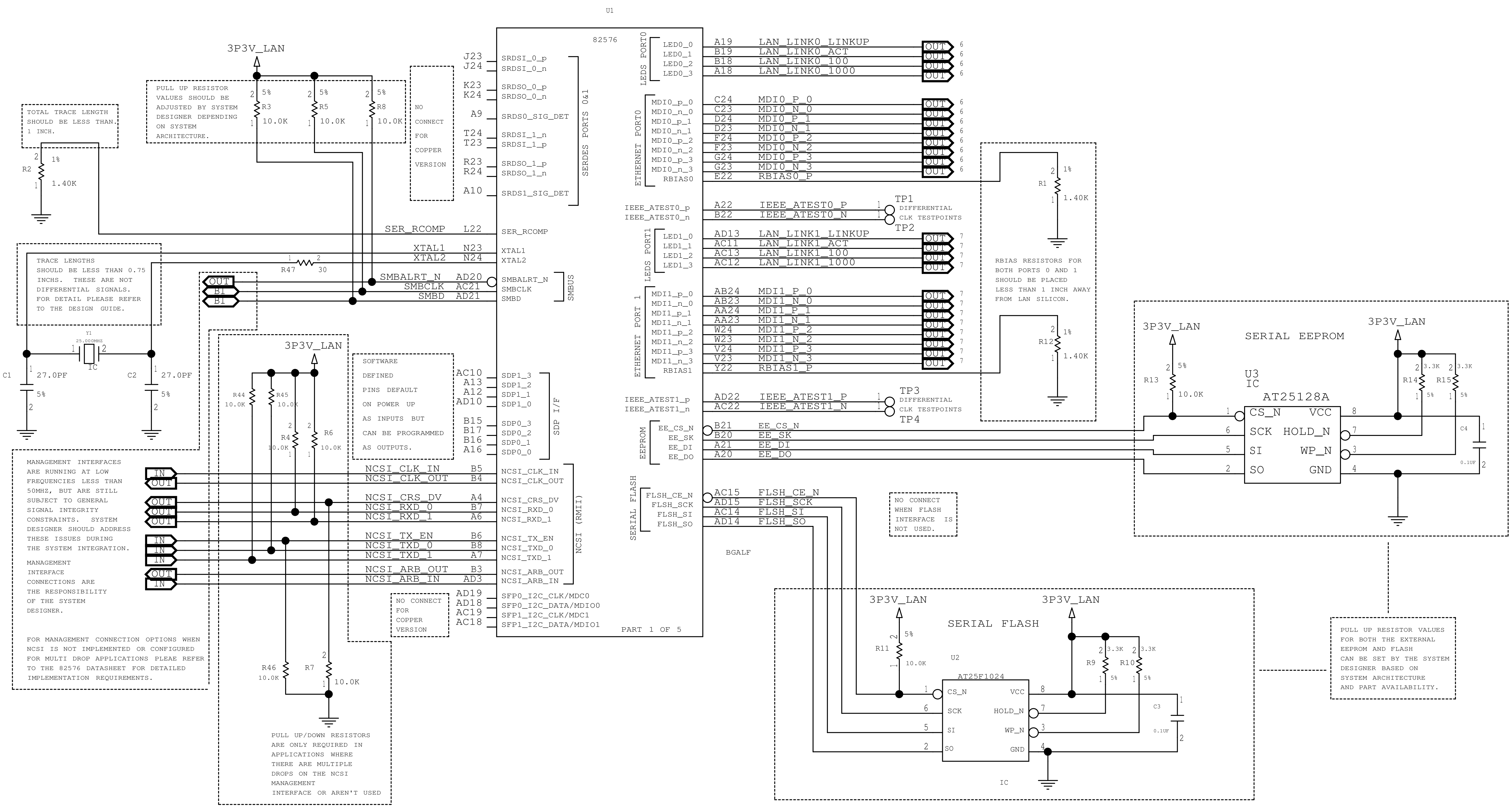
— SOLID LINE INDICATES THAT THE CONNECTIONS ARE SHOWN IN REFERENCE SCHEMATIC.

---- DASHED LINE INDICATES THAT THE CONNECTIONS ARE NOT SHOWN IN REFERENCE SCHEMATIC. THESE CONNECTIONS ARE OWNED BY THE SYSTEM DESIGNER.

2 - FUNCTIONAL BLOCK DIAGRAM

LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/15/2008	2



TOTAL TRACE LENGTH SHOULD BE LESS THAN. 1 INCH.

PULL UP RESISTOR VALUES SHOULD BE ADJUSTED BY SYSTEM DESIGNER DEPENDING ON SYSTEM ARCHITECTURE.

NO CONNECT FOR COPPER VERSION

TRACE LENGTHS SHOULD BE LESS THAN 0.75 INCHS. THESE ARE NOT DIFFERENTIAL SIGNALS. FOR DETAIL PLEASE REFER TO THE DESIGN GUIDE.

SOFTWARE DEFINED PINS DEFAULT ON POWER UP AS INPUTS BUT CAN BE PROGRAMMED AS OUTPUTS.

MANAGEMENT INTERFACES ARE RUNNING AT LOW FREQUENCIES LESS THAN 50MHZ, BUT ARE STILL SUBJECT TO GENERAL SIGNAL INTEGRITY CONSTRAINTS. SYSTEM DESIGNER SHOULD ADDRESS THESE ISSUES DURING THE SYSTEM INTEGRATION.

MANAGEMENT INTERFACE CONNECTIONS ARE THE RESPONSIBILITY OF THE SYSTEM DESIGNER.

FOR MANAGEMENT CONNECTION OPTIONS WHEN NCSI IS NOT IMPLEMENTED OR CONFIGURED FOR MULTI DROP APPLICATIONS PLEAE REFER TO THE 82576 DATASHEET FOR DETAILED IMPLEMENTATION REQUIREMENTS.

NO CONNECT FOR COPPER VERSION

PULL UP/DOWN RESISTORS ARE ONLY REQUIRED IN APPLICATIONS WHERE THERE ARE MULTIPLE DROPS ON THE NCSI MANAGEMENT INTERFACE OR AREN'T USED

RBIAS RESISTORS FOR BOTH PORTS 0 AND 1 SHOULD BE PLACED LESS THAN 1 INCH AWAY FROM LAN SILICON.

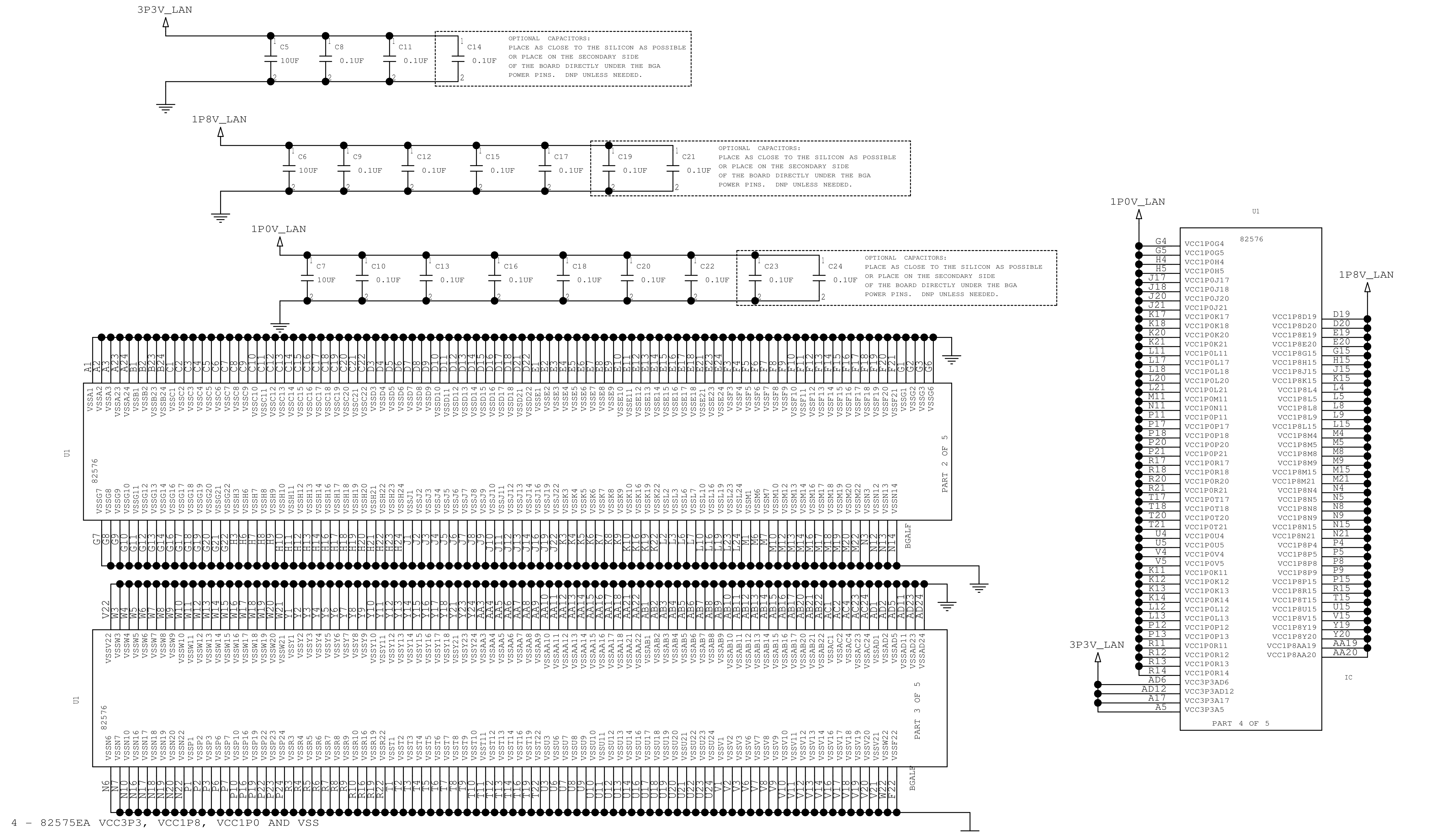
NO CONNECT WHEN FLASH INTERFACE IS NOT USED.

PULL UP RESISTOR VALUES FOR BOTH THE EXTERNAL EEPROM AND FLASH CAN BE SET BY THE SYSTEM DESIGNER BASED ON SYSTEM ARCHITECTURE AND PART AVAILABILITY.

3 - 82575EA MDI, LEDS, CRYSTAL, SERDES, SFP, NCSI, SMBUS, EEPROM AND FLASH

LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/15/2008	3



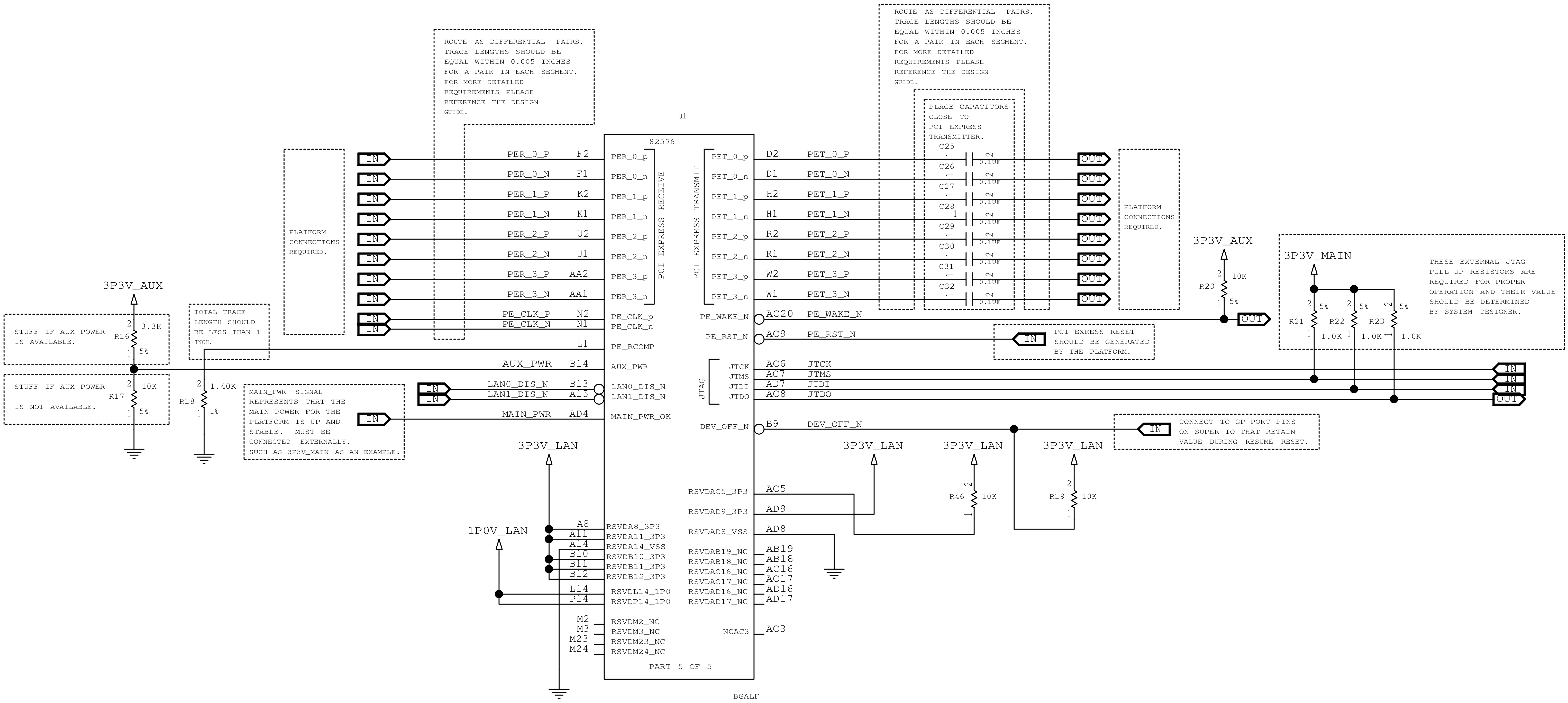
OPTIONAL CAPACITORS:
PLACE AS CLOSE TO THE SILICON AS POSSIBLE
OR PLACE ON THE SECONDARY SIDE
OF THE BOARD DIRECTLY UNDER THE BGA
POWER PINS. DNP UNLESS NEEDED.

OPTIONAL CAPACITORS:
PLACE AS CLOSE TO THE SILICON AS POSSIBLE
OR PLACE ON THE SECONDARY SIDE
OF THE BOARD DIRECTLY UNDER THE BGA
POWER PINS. DNP UNLESS NEEDED.

OPTIONAL CAPACITORS:
PLACE AS CLOSE TO THE SILICON AS POSSIBLE
OR PLACE ON THE SECONDARY SIDE
OF THE BOARD DIRECTLY UNDER THE BGA
POWER PINS. DNP UNLESS NEEDED.

4 - 82575EA VCC3P3, VCC1P8, VCC1P0 AND VSS

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000-000	REV 2.0	DATE 04/15/2008	SHEET 4



STUFF IF AUX POWER IS AVAILABLE.

STUFF IF AUX POWER IS NOT AVAILABLE.

TOTAL TRACE LENGTH SHOULD BE LESS THAN 1 INCH.

PLATFORM CONNECTIONS REQUIRED.

MAIN_PWR SIGNAL REPRESENTS THAT THE MAIN POWER FOR THE PLATFORM IS UP AND STABLE. MUST BE CONNECTED EXTERNALLY. SUCH AS 3P3V_MAIN AS AN EXAMPLE.

ROUTE AS DIFFERENTIAL PAIRS. TRACE LENGTHS SHOULD BE EQUAL WITHIN 0.005 INCHES FOR A PAIR IN EACH SEGMENT. FOR MORE DETAILED REQUIREMENTS PLEASE REFERENCE THE DESIGN GUIDE.

ROUTE AS DIFFERENTIAL PAIRS. TRACE LENGTHS SHOULD BE EQUAL WITHIN 0.005 INCHES FOR A PAIR IN EACH SEGMENT. FOR MORE DETAILED REQUIREMENTS PLEASE REFERENCE THE DESIGN GUIDE.

PLACE CAPACITORS CLOSE TO PCI EXPRESS TRANSMITTER.

PLATFORM CONNECTIONS REQUIRED.

THESE EXTERNAL JTAG PULL-UP RESISTORS ARE REQUIRED FOR PROPER OPERATION AND THEIR VALUE SHOULD BE DETERMINED BY SYSTEM DESIGNER.

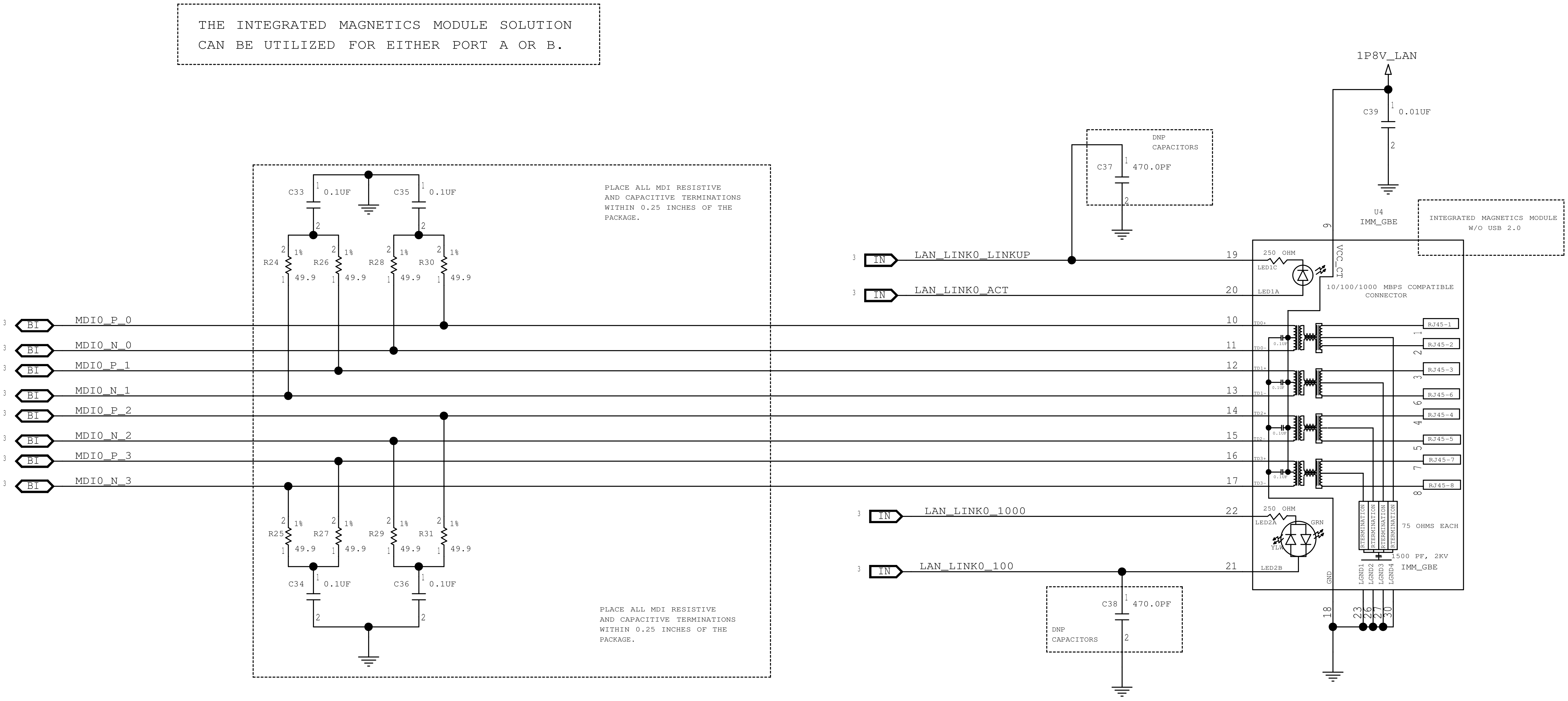
PCI EXRESS RESET SHOULD BE GENERATED BY THE PLATFORM.

CONNECT TO GP PORT PINS ON SUPER IO THAT RETAIN VALUE DURING RESUME RESET.

5 - 82575EA PCI EXPRESS, JTAG, AUX POWER, LAN DISABLE, MAIN POWER OK AND DEVICE OFF

LAN ACCESS DIVISION 2111 N.E. 25th AVENUE HILLSBORO, OR 97124	TITLE 82576 REF SCHEMATIC	SIZE B	CODE	DOCUMENT NUMBER 000000-000	REV 2.0	DATE 04/15/2008	SHEET 5

THE INTEGRATED MAGNETICS MODULE SOLUTION
CAN BE UTILIZED FOR EITHER PORT A OR B.

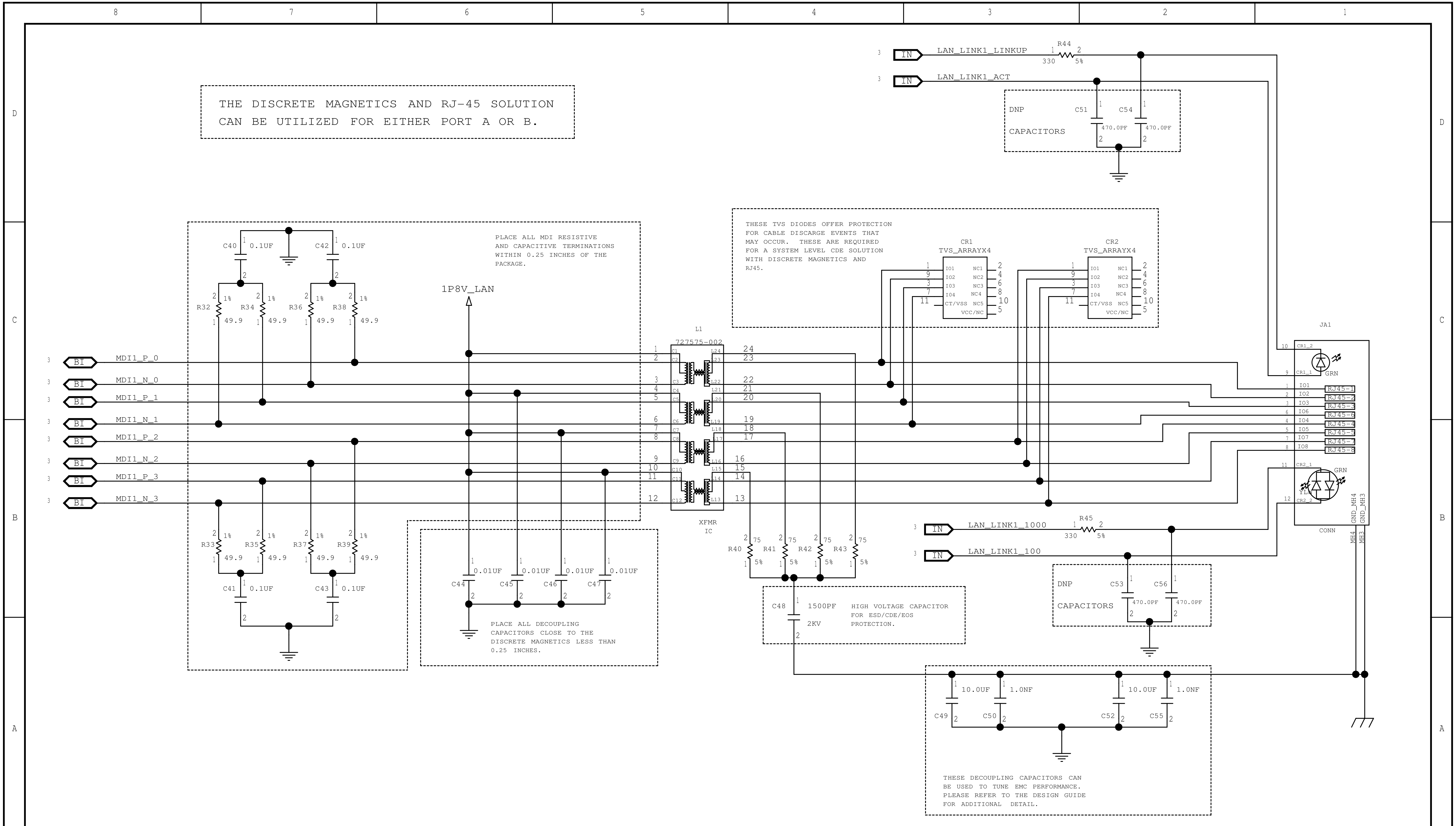


6 - ANALOG FRONT END OPTION 1 - INTEGRATED MAGNETICS MODULE W/O USB

LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/15/2008	6

THE DISCRETE MAGNETICS AND RJ-45 SOLUTION CAN BE UTILIZED FOR EITHER PORT A OR B.


LAN_LINK1_LINKUP
LAN_LINK1_ACT
R44 330 5%
DNP CAPACITORS
C51 470.0PF
C54 470.0PF
PLACE ALL MDI RESISTIVE AND CAPACITIVE TERMINATIONS WITHIN 0.25 INCHES OF THE PACKAGE.
C40 0.1UF
C42 0.1UF
R32 1% 49.9
R34 1% 49.9
R36 1% 49.9
R38 1% 49.9
1P8V_LAN
THESE TVS DIODES OFFER PROTECTION FOR CABLE DISCARGE EVENTS THAT MAY OCCUR. THESE ARE REQUIRED FOR A SYSTEM LEVEL CDE SOLUTION WITH DISCRETE MAGNETICS AND RJ45.
CR1 TVS_ARRAYX4
CR2 TVS_ARRAYX4
IO1 IO2 IO3 IO4 CT/VSS NC1 NC2 NC3 NC4 NC5 VCC/NC
JA1
BI
MDI1_P_0
MDI1_N_0
MDI1_P_1
MDI1_N_1
MDI1_P_2
MDI1_N_2
MDI1_P_3
MDI1_N_3
L1
727575-002
XFMR IC
GRN
YLW
RJ45-1 RJ45-2 RJ45-3 RJ45-6 RJ45-4 RJ45-5 RJ45-7 RJ45-8
IO1 IO2 IO3 IO6 IO4 IO5 IO7 IO8
CR1_1 CR1_2 CR2_1 CR2_2
GND_MH4 GND_MH3
MH4 MH3
CONN
R33 1% 49.9
R35 1% 49.9
R37 1% 49.9
R39 1% 49.9
C41 0.1UF
C43 0.1UF
C44 0.01UF
C45 0.01UF
C46 0.01UF
C47 0.01UF
PLACE ALL DECOUPLING CAPACITORS CLOSE TO THE DISCRETE MAGNETICS LESS THAN 0.25 INCHES.
R40 75 5%
R41 75 5%
R42 75 5%
R43 75 5%
C48 1500PF 2KV
HIGH VOLTAGE CAPACITOR FOR ESD/CDE/EOS PROTECTION.
IN
LAN_LINK1_1000
LAN_LINK1_100
R45 330 5%
DNP CAPACITORS
C53 470.0PF
C56 470.0PF
C49 10.0UF
C50 1.0NF
C52 10.0UF
C55 1.0NF
THESE DECOUPLING CAPACITORS CAN BE USED TO TUNE EMC PERFORMANCE. PLEASE REFER TO THE DESIGN GUIDE FOR ADDITIONAL DETAIL.

7 - ANALOG FRONT END OPTION 2 - DISCRETE MAGNETICS W/RJ45

LAN ACCESS DIVISION
2111 N.E. 25th AVENUE
HILLSBORO, OR 97124

TITLE	SIZE	CODE	DOCUMENT NUMBER	REV	DATE	SHEET
82576 REF SCHEMATIC	B		000000-000	2.0	04/15/2008	7