SWITCHING

Link Street® 88E6096/88E6097/ 88E6097F Datasheet

8 FE + 3 GE Stackable Ethernet Switch with QoS and 802.1Q

Doc. No. MV-S103922-00 Rev. B
July 2, 2007

MOVING FORWARD FASTER®

MARVELL®

Document Status	
Advance Information	This document contains design specifications for initial product development. Specifications may change without notice. Contact Marvell Field Application Engineers for more information.
Preliminary Information	This document contains preliminary data, and a revision of this document will be published at a later date. Specifications may change without notice. Contact Marvell Field Application Engineers for more information.
Final Information	This document contains specifications on a product that is in final release. Specifications may change without notice. Contact Marvell Field Application Engineers for more information.
Revision Code: Rev. B	
Advance	Technical Publication: 1.40

OVERVIEW

The Marvell® 88E6096, 88E6097, and 88E6097F devices are each single-chip 8-port 10/100 plus 3 port Gigabit Ethernet switches. These devices support 'Best-in-Class' Quality of Service (QoS) and the highest 'real-world' performance. They are uniquely suited for Small-to-Medium Business (SMB) and Access Switch applications.

These devices contain eight 10BASE-T/100BASE-TX transceivers (PHYs), and three gigabit SERDES interfaces that can be used to connect to external Marvell 10/100/1000 triple speed Ethernet transceivers (PHYs). The 88E6096/88E6097 devices are designed to work in all environments. True Plug-and-Play is supported with Auto-Crossover, Auto-Negotiation, and Auto-Polarity in the PHYs, along with bridge loop prevention (using Port States).

The 88E6096 is designed for SOHO and SMB switch applications, while the 88E6097 is for Access/Demarcation switch applications. The 88E6097F device is available for Access applications as well, and includes fiber enable and controls on all ports.

The devices offer enhanced stacking capabilities that enable multiple devices and/or systems to act as a single unit. These features include the ability to Link Aggregate ports between boxes in a stack, monitor any port from any other port in the stack, and create port-based VLANs using any port(s) within the stack.

The devices contain eleven independent 802.3 media access controllers (MACs), a high-speed, non-blocking four traffic class QoS switch fabric that uses the unique, patented Marvell Dynamic Queue Limit architecture. The QoS architecture switches packets into one of four traffic class queues based upon Port, IEEE 802.1p, IPv4 Type of Service (TOS) or Differentiated Services (Diff-Serv), IPv6 Traffic Class, 802.1Q VLAN ID, DA MAC address or SA MAC address. The devices also contain a high-performance address lookup engine with support for up to 8K active nodes, and a 1 Mbit frame buffer memory. Back-pressure and Pause frame-based flow control schemes are included to support zero packet loss under temporary traffic congestion. The MAC units in the devices comply fully with the applicable sections of IEEE 802.3.

The ninth and tenth ports' optional (G)MII (or MII) interface supports a direct connection to Management or Router CPUs with integrated MACs. The tenth port also supports an RGMII interface that operates as a 1000 Mbps, full-duplex port. For the 88E6096/88E6097 devices, Port 9 is only MII-PHY or MII-MAC mode capable, while Port 10 can be configured in either GMII, MII-PHY or MII-MAC mode. In the 88E6097F, Port 9 supports an additional GMII mode. These interfaces, along with BPDU handling for IEEE 802.1D Spanning Tree Protocol, 802.1w Rapid Spanning Tree, 802.1s Multiple VLAN Spanning Tree, programmable per-port VLAN configurations, 802.1Q and Port States, support fully managed switches and truly isolated WAN vs. LAN firewall applications. The devices support 4,096 802.1Q VLAN IDs which can be enabled on a per port basis. Three levels of 802.1Q security is supported with error frame trapping and logging.

The devices support multiple address databases (up to 4096), which allows packet routing without modification of the MAC address. This allows the same MAC address to exist multiple times in the MAC Address database with multiple port mappings, to completely isolate the WAN from the LAN database.

The PHY and SERDES units in the devices are designed with the Marvell® cutting-edge mixed-signal processing technology for digital implementation of adaptive equalization and clock data recovery. Special power management techniques are used to facilitate low power dissipation and high port count integration. Both the PHY and MAC units in the devices comply fully with the applicable sections of IEEE 802.3, IEEE 802.3u, and IEEE 802.3x standards.

The devices' many operating modes can be configured using SMI (serial management interface - MDC/MDIO) and/or a low cost serial EEPROM, and/or by special Remote Management ethernet frames. A standalone QoS mode is also supported.

Up to 32 88E6096/88E6097 devices can be cascaded. One or more devices can be used to create larger switches, e.g 24 FE + 2 GE or 48 FE + 4 GE. The devices are designed for cost-sensitive low to high port count switch systems that require Quality of Service, Trunking, Stacking, and/or Spanning Tree.

APPLICATIONS

- 8 FE + 2 1000BASE-X using a single 88E6096 or 88E6097 device
- 24 FE + 2 GE SMB or Access switch using three 88E6096 or three 88E6097 devices
- 48 FE + 4 GE SMB or Access switch using six 88E6096 or six 88E6097 devices with a single 88E6185, 98DX106, or 98DX107
- SOHO/SMB router with eight FE and two GE ports
- Fiber to the Curb cascade applications

Highlighted Features

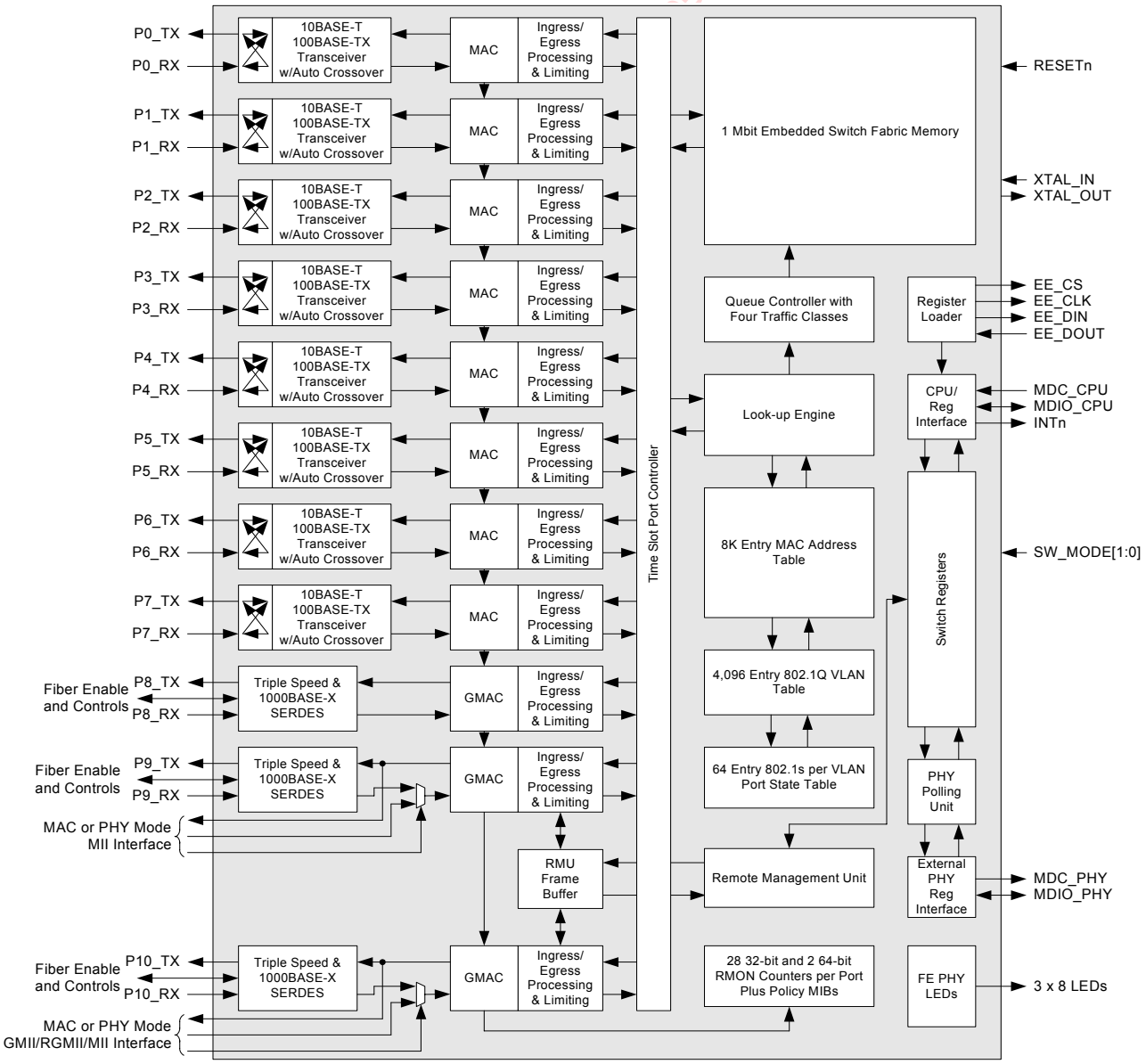
- 'Best-in-Class' per port TCP/IP Ingress Rate Limiting along with independent Storm Prevention
- 5 Ingress Rate Limiting buckets per port, supporting Rate-based and Priority-based rate limiting (88E6097 only)
- Non-Rate Limited frames based on SA or DA
- Layer 2 Policy Control List (PCL) enables drop, trap, or mirroring based on SA, DA, VID, Ethertype, VBAS, PPPoE, UDP, and DHCP Option 82 (88E6097 only)
- Remote Management capabilities allow device configuration and readback via Ethernet frames
- Per port, programmable MAC address limiting
- Quality of Service support with four traffic classes
- QoS determined by Port, IEEE 802.1p tagged frames, IPv4's Type of Service (TOS) & Differentiated Services (DS), IPv6's Traffic Class 802.1Q VID, Destination MAC address, or Source MAC address
- Frame priority overrides based on SA, DA, or VID
- Queue priority overrides based on SA, DA, VID, ARP, or Snoop
- Strict, Weighted, or mixed mode QoS selectable per port
- Globally Programmable QoS weighting via a 128-entry table (88E6097 only)
- 802.1Q VLAN support for the full 4,096 VLAN IDs
- Supports multiple provider ports within a single chip via a programmable Ethertype per port
- Enhanced 802.1s Per VLAN Spanning Tree supporting up to 64 spanning tree instances
- Support for protected port across multiple devices

Features

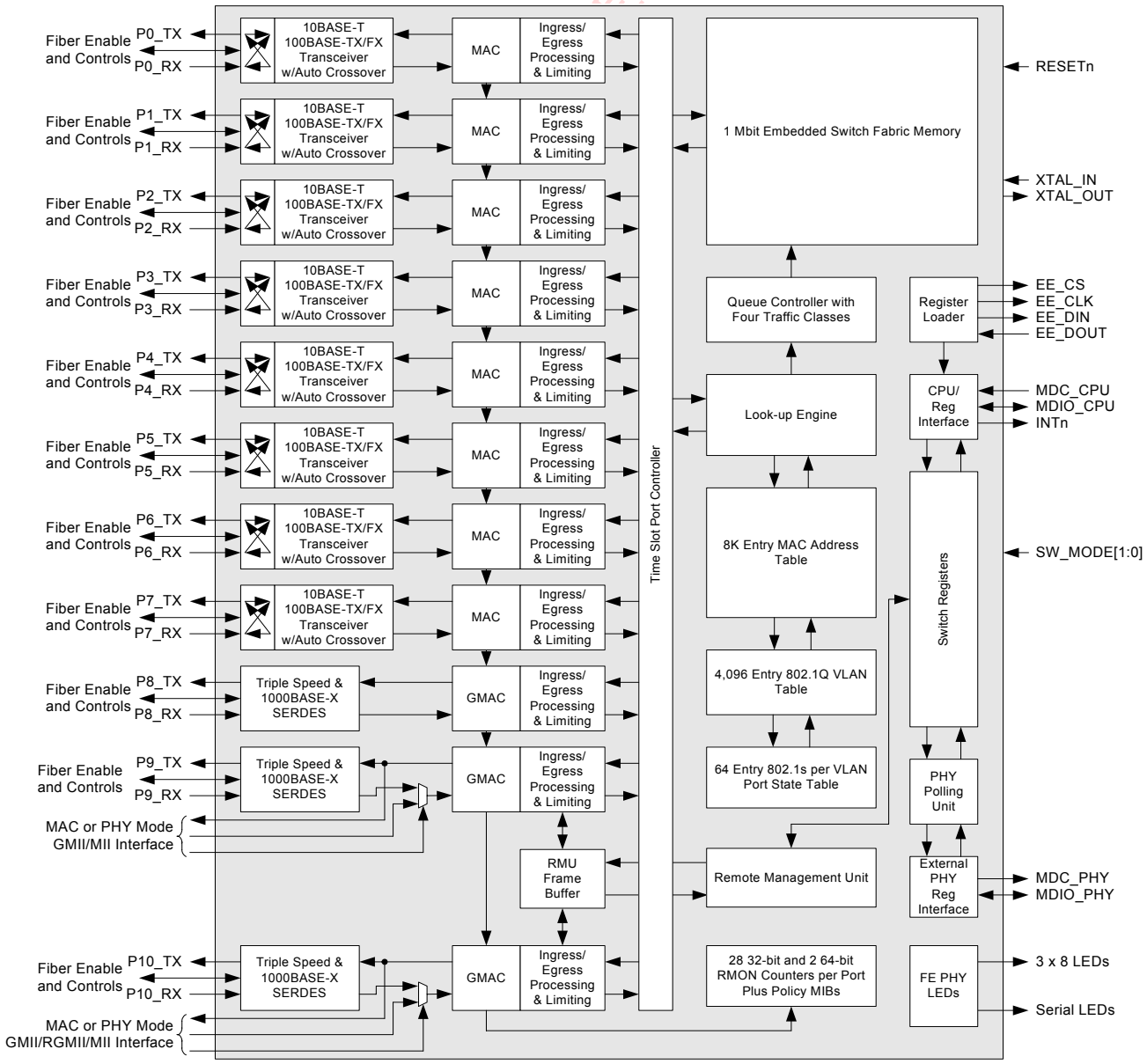
- Marvell® Header for increased Routing performance
- Shared 1 Mbit on-chip memory-based switch fabric with true non-blocking switching performance
- High performance lookup engine with support for up to 8K MAC address entries with automatic learning and aging
- Supports the Marvell Distributed Switching Architecture (DSA) for STP, up to 32 cascaded devices, and CPU-directed packet processing
- MAC SA based 802.1X authentication
- Port Trunking and Port Monitoring/Mirroring across chips
- Egress tagging/untagging selectable per port or by 802.1Q VLAN ID
- Port based VLANs supported in any combination across multiple chips
- Port States & BPDU handling for Spanning Tree
- 28 32-bit and 2 64-bit RMON Counters per port
- Ports 8, 9 & 10 are independent triple speed SERDES transceivers to interface with Marvell Alaska® gigabit copper PHYs and can optionally be configured as fiber ports (1000BASE-X) with direct connection to lasers
- Ports 9 and 10 can support GMII Mode (full-duplex), MII-MAC Mode (Forward) or MII-PHY Mode (Reverse—full-duplex) interface options for management and firewall applications (Port 9 is MII only in the 88E6096/88E6097 devices)
- Port 10 supports an additional RGMII interface that operates at 1000 Mbps, full-duplex only
- Integrated with eight independent Auto-Crossover Fast Ethernet transceivers fully compliant with the applicable sections of IEEE802.3 and IEEE802.3u
- Flexible LED support for Link, Speed, Duplex Mode, Collision, and Tx/Rx Activities
- Supports a low-cost 25 MHz XTAL clock source
- Supports 4-Wire 93C56/93C66 or 2-Wire 24C01/24C02/24C04 EEPROMs
- Single chip integration of an 11 port 8-FE + 3-GE QoS switch and memory in a 20 x 20 mm 176-pin TQFP package (88E6096/88E6097 device), or 24 x 24 mm 216 LQFP (88E6097F device)
- Pin compatible to the previous generation 88E6092 and 88E6095 devices
- 88E6097 and 88E6097F devices available in Commercial and Industrial grade
- Low power dissipation P_{AVE} = 1.5W

Table 1: 88E6096/88E6097/88E6097F Device Differences

	Description	88E6096 SOHO/SMB Switches	88E6097 Access Switches	88E6097F Access Switches
Features	FE PHY and GE SERDES	8 + 3	8 + 3	8+3
	FE Fiber Support	0	0	8
	# GMII/MII (Shared w/SERDES)	1/2	1/2	2/2
	RGMII	1	1	1
	Max Frame Size	1632 bytes	1632 bytes	1632 bytes
	Packet Buffer Memory	1 Mbit	1 Mbit	1 Mbit
	# MAC Addresses	8K	8K	8K
QoS	Queues per Port	4	4	4
	802.1p, Port, TOS/DS, IPv6, TC, MAC	Yes	Yes	Yes
	Priority Overrides (DA/SA/VID/ARP/Snoop)	Yes	Yes	Yes
	Mixed Mode QoS per port	Yes	Yes	Yes
	Programmable Weighting	No	Yes	Yes
VLAN	Port -based and 802.1Q VLANs	Yes	Yes	Yes
	Maximum number of Shared 802.1Q VLANs	4096	4096	4096
	Maximum number of Independent 802.1Q VLANs	4096	4096	4096
	Double Tagging (Q in Q)	Yes	Yes	Yes
	Supports Multiple Provider Ports	Yes	Yes	Yes
Cross-chip	Stacking Support	Yes	Yes	Yes
	Cross -chip Trunking (#Trunks/#Ports)	Yes (16/8)	Yes (16/8)	Yes (16/8)
	Cross-chip Port Based VLANs	Yes	Yes	Yes
	Cross-chip Flow Control	Yes	Yes	Yes
	Cross-chip Protected Port	Yes	Yes	Yes
	Distributed Switching Architecture (DSA) Tag Support	Yes	Yes	Yes
Management	Remote Management	Yes	Yes	Yes
	Ethertype DSA	Yes	Yes	Yes
	Layer 2 Policy Control Lists (PCL)	No	Yes	Yes
	802.1D/w/s Spanning Tree	Yes	Yes	Yes
	Port Mirroring	Yes	Yes	Yes
	IPv4 IGMP & IPv6 MLD Snooping	Yes	Yes	Yes
Other	Ingress Rate Limiting Resources	2/port	5/port	5/port
	Priority-based Ingress Rate Limiting	No	Yes	Yes
	Egress Rate Shaping	Yes	Yes	Yes
	802.1X Port and MAC-based Authentication	Yes	Yes	Yes
	Hardware MAC Address Limiting per port	Yes	Yes	Yes
	Industrial Grade	No	Yes	Yes
	Package	176-pin LQFP	176-pin LQFP	216-pin LQFP



88E6096/88E6097 176 TQFP Top Level Block Diagram



88E6097F 216 LQFP Top Level Block Diagram

Table of Contents

List of Tables

List of Figures

Section 1. Signal Description

Figure 1: 88E6096/88E6097 Device 176 TQFP EPAD Package Pinout

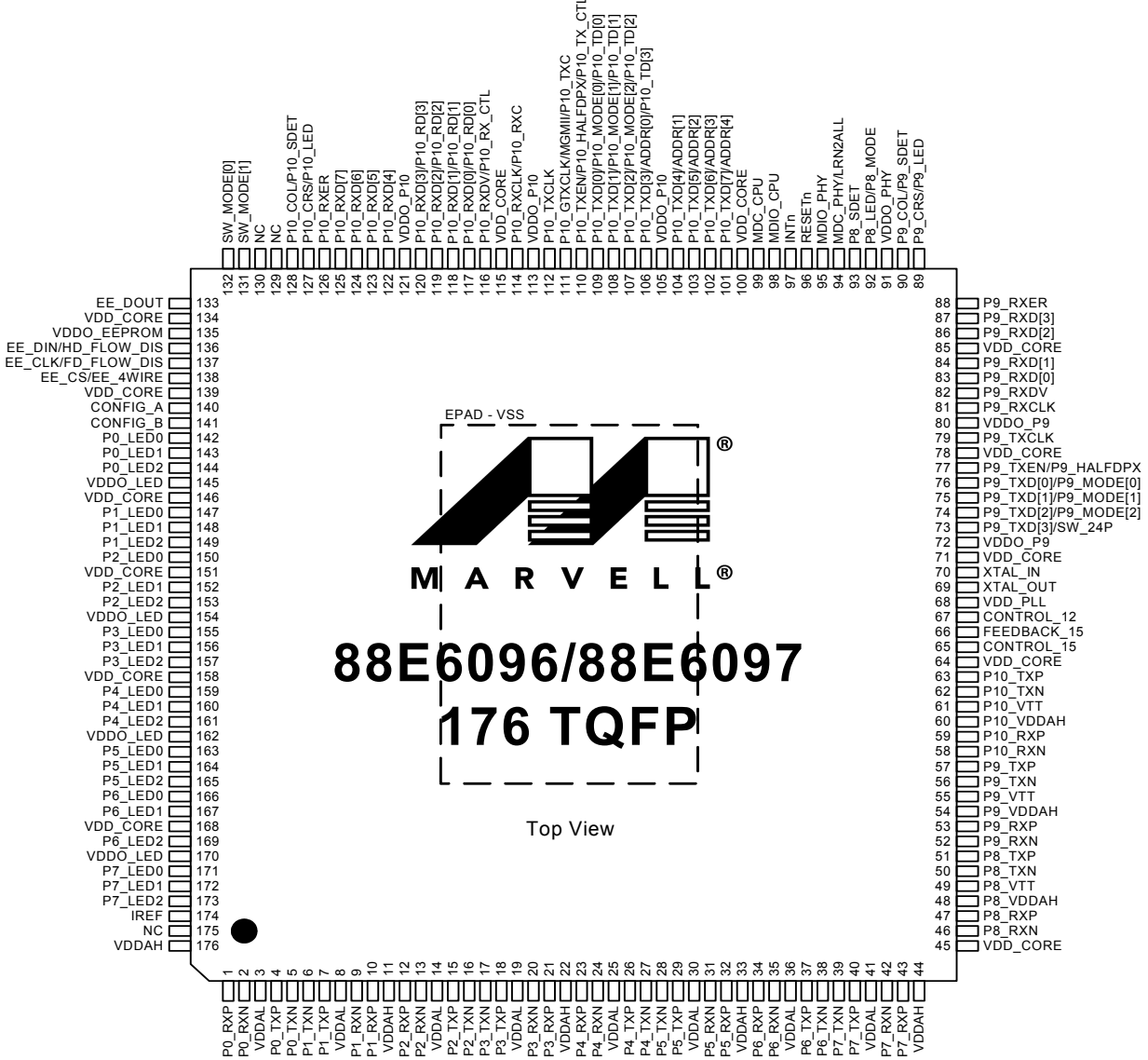
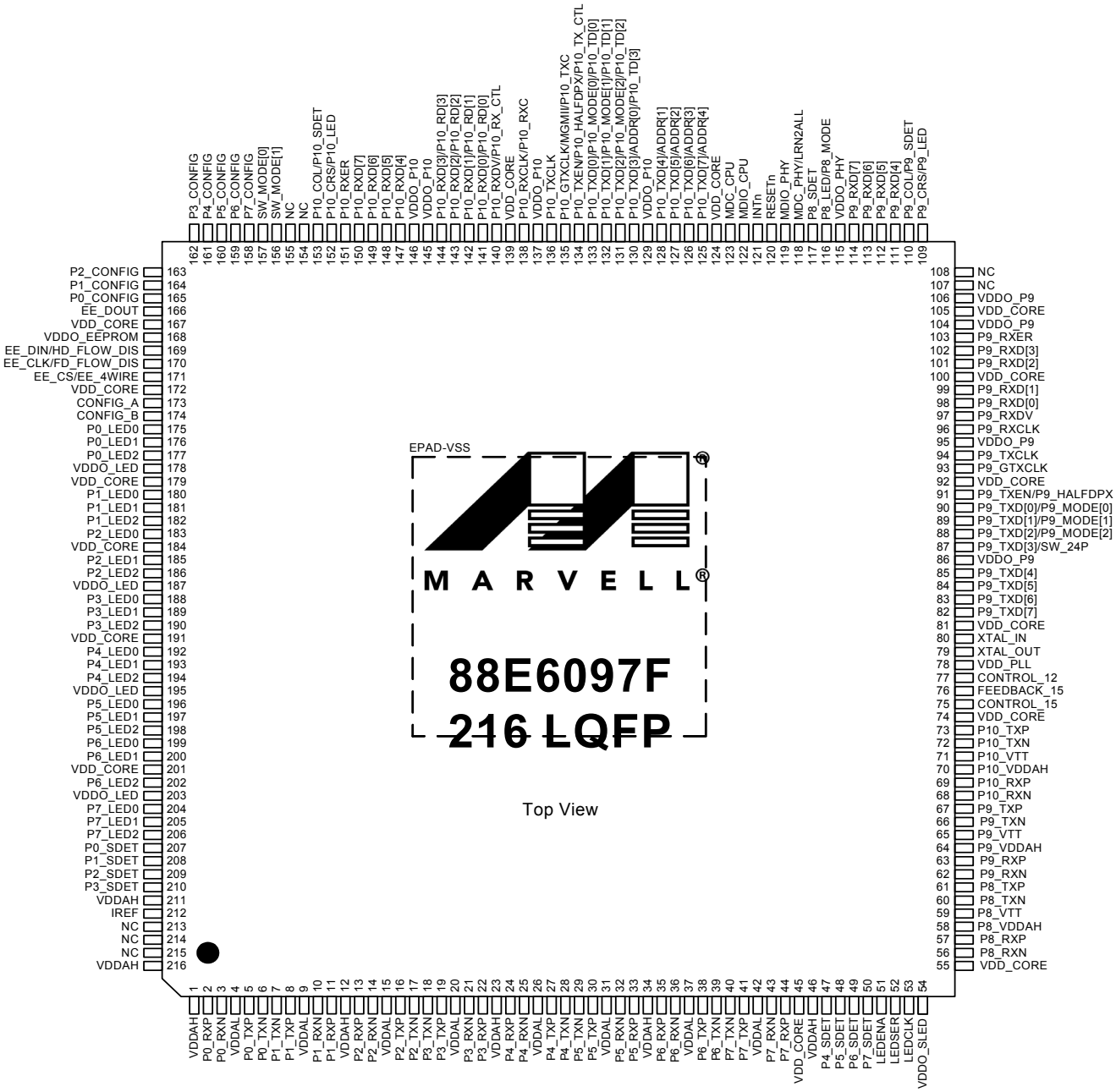


Figure 2: 88E6097F Device 216 LQFP EPAD Package Pinout



1.1 Pin Description

Table 2: Pin Type Definitions

Pin Type	Definition
H	Input with hysteresis
I/O	Input and output
I	Input only
O	Output only
PU	Internal pull-up
PD	Internal pull-down
D	Open drain output
Z	Tri-state output
mA	DC sink capability
Analog	Analog pin

Table 3: Network 10/100 PHY Interface (Ports 0 to 7)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
43 34 32 23 21 12 10 1	44 35 33 24 22 13 11 2	P7_RXP P6_RXP P5_RXP P4_RXP P3_RXP P2_RXP P1_RXP P0_RXP	Typically Input	Receiver input – Positive. P[7:0]_RXP connects directly to the receiver magnetics. If the port is configured for 100BASE-FX mode (available in the 88E6097F device only) RXP connects directly to the fiber-optic receiver's positive output. For lowest power, all unused port RXP pins should be tied to VSS. These pins can become outputs if Auto MDI/MDIX Crossover is enabled.
42 35 31 24 20 13 9 2	43 36 32 25 21 14 10 3	P7_RXN P6_RXN P5_RXN P4_RXN P3_RXN P2_RXN P1_RXN P0_RXN	Typically Input	Receiver input – Negative. P[7:0]_RXN connects directly to the receiver magnetics. If the port is configured for 100BASE-FX mode (available in the 88E6097F device only) RXN connects directly to the fiber-optic receiver's negative output. For lowest power, all unused port RXN pins should be tied to VSS. These pins can become outputs if Auto MDI/MDIX Crossover is enabled.
40 37 29 26 18 15 7 4	41 38 30 27 19 16 8 5	P7_TXP P6_TXP P5_TXP P4_TXP P3_TXP P2_TXP P1_TXP P0_TXP	Typically Output	Transmitter output – Positive. P[7:0]_TXP connects directly to the transmitter magnetics. If the port is configured for 100BASE-FX mode (available in the 88E6097F device only) TXP connects directly to the fiber-optic transmitter's positive input. For lowest power, all unused port TXP pins should be tied to VSS. These pins can become inputs if Auto MDI/MDIX Crossover is enabled.
39 38 28 27 17 16 6 5	40 39 29 28 18 17 7 6	P7_TXN P6_TXN P5_TXN P4_TXN P3_TXN P2_TXN P1_TXN P0_TXN	Typically Output	Transmitter output – Negative. P[7:0]_TXN connects directly to the transmitter magnetics. If the port is configured for 100BASE-FX mode (available in the 88E6097F device only) TXN connects directly to the fiber-optic transmitter's negative input. For lowest power, all unused port TXN pins should be tied to VSS. These pins can become inputs if Auto MDI/MDIX Crossover is enabled.
	50 49 48 47 210 209 208 207	P7_SDET P6_SDET P5_SDET P4_SDET P3_SDET P2_SDET P1_SDET P0_SDET	Input	Signal Detect input (88E6097F device only). If port 0 to 7 are configured for 100BASE-FX mode Px_SDET indicates whether a signal is detected by the fiber-optic transceiver. A positive level indicates that a signal is detected. If port 0 to 7 are configured for 10/100BASE-T mode Px_SDET is not used but it can not be left floating since these pins do not contain internal resistors. Px_SDET can be tied to VSS or VDDO either directly or through a 4.7kΩ resistor.

Table 4: PHY Configuration (Ports 0 to 7)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
	158 159 160 161 162 163 164 165	P7_CONFIG P6_CONFIG P5_CONFIG P4_CONFIG P3_CONFIG P2_CONFIG P1_CONFIG P0_CONFIG	Input	Port 7 to 0 Configuration (88E6097F device only). The Px_CONFIG pin is used to set the default configuration for Port 7 to 0 by connecting these pins to other device pins as follows: VSS = Auto-Negotiation enabled P0_LED1 = Forced 10BASE-T half-duplex P0_LED2 = Forced 10BASE-T full-duplex P1_LED0 = Forced 100BASE-TX half-duplex P1_LED1 = Forced 100BASE-TX full-duplex P1_LED2 = Forced 100BASE-FX half-duplex VDDO = Forced 100BASE-FX full-duplex Ports 7 to 0 default configuration is Auto-Negotiation enabled in the 88E6097F device. Any port's default configuration can be modified by accessing the PHY registers by a CPU or a serial EEPROM. Fiber mode vs. copper mode cannot be configured in this way, however. Fiber vs. copper must be selected at Reset by using these pins. Px_CONFIG pins are configured after reset and contain internal pull-down resistors so they can be left floating. **NOTE:** For the 88E6096/88E6097, the port's default configuration is Auto-Negotiation enabled.
140	173	CONFIG_A	Input	Global Configuration A. This global configuration pin is used to set the default LED mode and Far End Fault Indication (FEFI) mode by connecting these pins to other device pins as follows: VSS = LED Mode 0, FEFI disabled P0_LED0 = LED Mode 0, FEFI enabled P0_LED1 = LED Mode 1, FEFI disabled P0_LED2 = LED Mode 1, FEFI enabled P1_LED0 = LED Mode 2, FEFI disabled P1_LED1 = LED Mode 2, FEFI enabled P1_LED2 = LED Mode 3, FEFI disabled VDDO = LED Mode 3, FEFI enabled The CONFIG_A pin is configured after reset and does contain an internal pull-up resistor so it can be left floating.

Table 4: PHY Configuration (Ports 0 to 7) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
141	174	CONFIG_B	Input	Global Configuration B. This global configuration pin is used to set the default mode for Auto Crossover, the PHY driver type and Energy Detect by connecting these pins to other device pins as follows: VSS = No Crossover, Class A drivers, Energy Detect disabled P0_LED0 = No Crossover, Class A drivers, Energy Detect enabled P0_LED1 = No Crossover, Class B drivers, Energy Detect disabled P0_LED2 = No Crossover, Class B drivers, Energy Detect enabled P1_LED0 = Auto Crossover, Class A drivers, Energy Detect disabled P1_LED1 = Auto Crossover, Class A drivers, Energy Detect enabled P1_LED2 = Auto Crossover, Class B drivers, Energy Detect disabled VDDO = Auto Crossover, Class B drivers, Energy Detect enabled The CONFIG_B pin is configured after reset and does contain an internal pull-up resistor so it can be left floating.

Table 5: Port Status LEDs (Ports 0 to 7)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
173 169 165 161 157 153 149 144	206 202 198 194 190 186 182 177	P7_LED2 P6_LED2 P5_LED2 P4_LED2 P3_LED2 P2_LED2 P1_LED2 P0_LED2	Output	Parallel LED outputs—one for each port. This active low LED pin directly drives an led in parallel led mode. it can be configured to display many options. P[7:0]_LED2 are driven active low whenever RESETn is active low.
172 167 164 160 156 152 148 143	205 200 197 193 189 185 181 176	P7_LED1 P6_LED1 P5_LED1 P4_LED1 P3_LED1 P2_LED1 P1_LED1 P0_LED1	Output	Parallel LED outputs – one for each port. This active low LED pin directly drives an LED in Parallel LED mode. It can be configured to display many options. P[7:0]_LED1 are driven active low whenever RESETn is active low.
171 166 163 159 155 150 147 142	204 199 196 192 188 183 180 175	P7_LED0 P6_LED0 P5_LED0 P4_LED0 P3_LED0 P2_LED0 P1_LED0 P0_LED0	Output	Parallel LED outputs – one for each port. This active low LED pin directly drives an LED in Parallel LED mode. It can be configured to display many options. P[7:0]_LED0 are driven active low whenever RESETn is active low.
	52	LEDSER	Output	88E6097F device only; LEDSER outputs serial status bits that can be shifted into a shift register to be displayed via LEDs. LEDSER is output synchronously with LEDCLK.
	51	LEDENA	Output	88E6097F device only; LEDENA asserts High whenever LEDSER has valid status that is to be stored into the shift register. LEDENA is output synchronously with LEDCLK.
	53	LEDCLK	Output	88E6097F device only; LEDCLK is the reference clock for the serial LED signals.

Table 6: SERDES/1000BASE-X Interface (Ports 8 to 10)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
59 53 47	69 63 57	P10_RXP P9_RXP P8_RXP	Input	Receiver input – Positive. P[10:8]_RXP connects directly to the fiber-optic receiver's positive output or to another device's TXP (Transmitter output—positive) pins.
58 52 46	68 62 56	P10_RXN P9_RXN P8_RXN	Input	Receiver input – Negative. P[10:8]_RXN connects directly to the fiber-optic receiver's negative output or to another device's TXN (Transmitter output—negative) pins.
63 57 51	73 67 61	P10_TXP P9_TXP P8_TXP	Output	Transmitter output – Positive. P[10:8]_TXP connects directly to the fiber-optic transmitter's positive input or to another device's RXP (Receiver input—positive) pins.
62 56 50	72 66 60	P10_TXN P9_TXN P8_TXN	Output	Transmitter output – Negative. P[10:8]_TXN connects directly to the fiber-optic transmitter's negative input or to another device's RXN (Receiver input – Negative) pins.
93	117	P8_SDET	Input	Signal Detect input. If Port 8 is configured for 1000BASE-X mode (see Px_MODE below) Px_SDET is an input that is used to indicate whether a signal is detected by the fiber-optic transceiver. A positive level indicates that a signal is detected. If Port 8 is configured for SERDES mode P8_SDET is not used but it can be left floating since this pin contains an internal pull-down resistor.

Table 6: SERDES/1000BASE-X Interface (Ports 8 to 10) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
92	116	P8_LED /P8_MODE	Typically Output	LED Output and Mode. If Port 8 is configured for 1000BASE-X mode Px_LED indicates whether a 'Link' signal is detected by the Px_SDET input. A low output level indicates that a positive input signal is detected by Px_SDET and the port's PCS has determined link. Px_LED indicates 'Activity' by blinking high when the port receives or transmits a frame. Px_LED are multi-function pins used to configure the device during a hardware reset. When reset is asserted, Px_LED become inputs and the Port's configuration is latched at the rising edge of RESETn as follows: Low = Port is configured for cross-chip SERDES mode, AutoNeg is off High = Port is configured for 1000BASE-X mode AutoNeg is on **NOTE:** MGMII mode overrides the port's configuration made by this pin. MGMII mode is selected when the PHY Polling Unit (PPU) detects an external PHY is connected to this port. See "P10_GTXCLK/MGMII" pin description in Table 10, "GMII/MII Transmit Interface (Port 10)," on page 39 for MGMII mode details. Px_LED is internally pulled low via a resistor so the pin can be left floating when unused or to select cross-chip SERDES mode. Connecting an external LED and its resistor to VDDO will configure the port for 1000BASE-X mode.

Table 7: GMII/MII Receive Interface (Port 9 is MII only - 88E6096/88E6097 devices only)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
81	96	P9_RXCLK	Input	Receive Clock. RXCLK is a reference for RXDV, RX_ER and RXD[7:0]. The speed of RXCLK is expected to be 125 MHz, 25 MHz or 2.5 MHz depending upon the speed of the port. P9_RXCLK is internally pulled high so the pin can be left unconnected if not used.
    87 86 84 83	114 113 112 111 102 101 99 98	P9_RXD[7] P9_RXD[6] P9_RXD[5] P9_RXD[4] P9_RXD[3] P9_RXD[2] P9_RXD[1] P9_RXD[0]	Input	P9_RXD[7:0]—88E6097F device Only. P9_RXD[3:0]—88E6096/88E6097 devices Only. Receive Data RXD[7:0] receives the data octet or nibble to be sent into the switch. RXD[7:0] must be synchronous to RXCLK. In 1000BASE mode, RXD[7:0] is used. In 100BASE and 10BASE modes, RXD[3:0] is used and RXD[7:4] is ignored. P9_RXD[7:0] are internally pulled low via resistor so the pins can be left unconnected when they are not used.
82	97	P9_RXDV	Input	Receive Data Valid. Receive Data Valid is used to indicate when RXD[7:0] (or RXD[3:0] where appropriate) contains frame information. RXDV must be synchronous to RXCLK. P9_RXDV is internally pulled low via resistor so the pin can be left unconnected when it is not used.
88	103	P9_RXER	Input	Receive Data Error. Receive Data Error is used to indicate when an external device (like a PHY) detects an error. When this signal is high, the receive MAC will discard the frame and it will be counted as a frame with a CRC error. RXER must be synchronous to RXCLK. P9_RXER is internally pulled low via resistor so the pin can be left unconnected when it is not used.

Table 7: GMII/MII Receive Interface (Port 9 is MII only - 88E6096/88E6097 devices only) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
89	109	P9_CRS /P9_LED	Input or Output	Carrier Sense. Carrier sense is used to indicate carrier has been detected on the line. CRS is not synchronous to RXCLK. CRS is used for half-duplex modes only and is ignored when the port is in full-duplex. When Port 9 is configured for 1000BASE-X operation (see P9_MODE[2:0]) Px_LED indicates whether a 'Link' signal is detected by the Px_SDET input. A low output level indicates that a positive input signal is detected by Px_SDET and the port's PCS has determined link. Px_LED indicates 'Activity' by blinking high when the port receives or transmits a frame. CRS is internally pulled low via resistor so the pin can be left unconnected when it is not used. The direction of this pin is determined by the value of the P9_MODE[2:0] pins at the rising edge of RESETn (during RESETn this pin is an input).
90	110	P9_COL /P9_SDET	Input	Collision. Collision is used to indicate both transmit and receive are occurring at the same time in half-duplex mode. The Collision signal is not synchronous to RXCLK. The Collision signal is used for half-duplex modes only and is ignored when the port is in full-duplex. When Port 9 is configured for 1000BASE-X operation (see P9_MODE[2:0]) Px_SDET is an input used to indicate whether a signal is detected by the fiber-optic transceiver. A positive level indicates that a signal is detected. P9_COL is internally pulled low via resistor so the pin can be left unconnected when it is not used.

Table 8: GMII/MII Transmit Interface (Port 9)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
	93	P9_GTXCLK	Output	Transmit Clock; 88E6097F device only; GTXCLK is a reference for TXEN and TXD[7:0] when the port is in GMII mode. The speed of GTXCLK is 125 MHz and is normally only driven when the speed of the port is 1000BASE. GTXCLK can be configured to output a 25 MHz or 2.5 MHz clock so it can be used as a clock source for P9_TXCLK and P9_RXCLK when no other clock sources are available (see P9_MODE[2:0]). P9_GTXCLK is tri-stated during RESETn and it is internally pulled high so the pin can be left unconnected if not used.
79	94	P9_TXCLK	Input	Transmit Clock. TXCLK is a reference for TXEN and TXD[3:0] when the port is in MII mode. The speed of TXCLK is 25 MHz or 2.5 MHz depending the speed of the port. P9_TXCLK is tri-stated during RESETn and it is internally pulled high so the pin can be left unconnected if not used.

Table 8: GMII/MII Transmit Interface (Port 9) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
	82 83 84 85	P9_TXD[7] P9_TXD[6] P9_TXD[5] P9_TXD[4]	Typically Output	P9_TXD[7:0]/P9_MODE[2:0]—88E6097F device Only P9_TXD[3:0]/P9_MODE[2:0]—88E6096/88E6097 devices Only Transmit Data. TXD[7:0] outputs the data octet to be transmitted from the switch. TXD[7:0] is synchronous to GTXCLK in 1000BASE mode. In 100BASE and 10BASE modes TXD[3:0] is synchronous to TXCLK and TXD[7:4] is ignored. P9_TXD[7:0] are multi-function pins used to configure the device during a hardware reset. When reset is asserted, these pins become inputs and the configuration information below is latched at the rising edge of RESETn: TXD[2:0] = P9_MODE[2:0] TXD[3] = SW_24P P9_MODE[2:0] sets Port 9's Mode of operation as follows: 000 = GMII with P9_GTXCLK = 125 MHz (1000BASE)[1] 001 = MII with P9_GTXCLK = 0 MHz (MII MAC Mode) 010 = MII (PHY Mode) - Port 9 MII PHY mode must source 25 MHz or 2.5 MHz clock from P10_GTXCLK or external clock 011 = MII (PHY Mode)- Port 9 MII PHY mode must source 25 MHz or 2.5 MHz clock from P10_GTXCLK or external clock 100 = Cross-chip SERDES port with all GMII/MII pins tri-stated - This mode configures the SERDES for 1000BASE-X operation, with Auto-Negotiation disabled. This mode is recommended for use when cascading multiple switch devices and is not recommended when connecting to PHY devices. 101 = 1000BASE-X with all GMII/MII pins tri-stated (except for P9_LED) 110 = SGMII or MGMII with all GMII/MII pins tri-stated. See "P10_GTXCLK/MGMII" pin description in Table 10, "GMII/MII Transmit Interface (Port 10)," on page 39 for MGMII mode details. 111 = Port 9 is disabled (with all GMII/MII pins tri-stated)
73	87	P9_TXD[3] /SW_24P		
74	88	P9_TXD[2] /P9_MODE[2]		
75	89	P9_TXD[1] /P9_MODE[1]		
76	90	P9_TXD[0] /P9_MODE[0]		

Table 8: GMII/MII Transmit Interface (Port 9) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
(Cont. from table above)	(Cont. from table above)	(Cont. from table above)	Typically Output	SW_24P configures the device for cascading mode: • Cross-chip SERDES on all Ports 8, 9, and 10 interfaces, • Separate PortBased VLAN between Port 8 and Port 9 • Separate VLAN of Port 10 for CPU • Enabled configuration of Learn2All - see Learn2All section P9_TXD[7:0] are internally pulled low via resistor so the pins can be left unconnected when they are not used. Use a 4.7 kohm resistor to VDDO for a configuration high.
77	91	P9_TXEN /P9_HALFDPX	Typically Output	Transmit Enable. Transmit enable is used to indicate when TXD[7:0] (or TDX[3:0] where appropriate) contains frame information. TXEN is synchronous to GTXCLK in 1000BASE mode and its synchronous to TXCLK in 100BASE and 10BASE modes. P9_TXEN is a multi-function pin used to configure the device during a hardware reset. When reset is asserted, this pin become an input and the configuration information below is latched at the rising edge of RESETn: 0 = Sets Port 9 in full-duplex operation 1 = Sets Port 9 in half-duplex operation only if P9_MODE[2:0] = 01X P9_TXEN is internally pulled low via resistor so the pin can be left unconnected to select full-duplex on Port 9 or when it is not used.

1. P9_MODE[2:0] = 000 is not supported in the 88E6096/88E6097 devices

Table 9: GMII/MII Receive Interface (Port 10)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
114	138	P10_RXCLK	Input	Receive Clock. RXCLK is a reference for RXDV, RX_ER and RXD[7:0]. The speed of RXCLK is expected to be 125 MHz, 25 MHz or 2.5 MHz depending upon the speed of the port. P10_RXCLK is tri-stated during RESETn and it is internally pulled high so the pin can be left unconnected if not used. **NOTE:** This pin can also be used as the RGMII Receive Clock pin as documented in Table 11.
125 124 123 122 120 119 118 117	150 149 148 147 144 143 142 141	P10_RXD[7] P10_RXD[6] P10_RXD[5] P10_RXD[4] P10_RXD[3] P10_RXD[2] P10_RXD[1] P10_RXD[0]	Input	Receive Data. RXD[7:0] receives the data octet or nibble to be sent into the switch. RXD[7:0] must be synchronous with RXCLK. In 1000BASE mode, RXD[7:0] is used. In 100BASE and 10BASE modes, RXD[3:0] is used and RXD[7:4] is ignored. P10_RXD[7:0] are internally pulled low via resistor so the pins can be left unconnected when they are not used. **NOTE:** P10_RXD[3:0] can also be used as the RGMII Receive Data pins as documented in Table 11.
116	140	P10_RXDV	Input	Receive Data Valid. Receive Data Valid is used to indicate when RXD[7:0] (or RXD[3:0] where appropriate) contains frame information. RXDV must be synchronous to RXCLK. P10_RXDV is internally pulled low via resistor so the pin can be left unconnected when it is not used. **NOTE:** This pin can also be used as the RGMII Receive Control pin as documented in Table 11.
126	151	P10_RXER	Input	Receive Data Error. Receive Data Error is used to indicate when an external device (like a PHY) detects an error. When this signal is high the receive MAC will discard the frame and it will be counted as a frame with a CRC error. RXER must be synchronous to RXCLK. P10_RXER is internally pulled low via resistor so the pin can be left unconnected when it is not used.

Table 9: GMII/MII Receive Interface (Port 10) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
127	152	P10_CRS /P10_LED	Input or Output	Carrier Sense. Carrier sense is used to indicate that carrier has been detected on the line. CRS is not synchronous to RXCLK. CRS is used for half-duplex modes only and is ignored when the port is in full-duplex. When Port 10 is configured for 1000BASE-X operation (see P9/P10_MODE[2:0]) Px_LED indicates whether a 'Link' signal is detected by the Px_SDET input. A low output level indicates that a positive input signal is detected by Px_SDET and the port's PCS determines link. Px_LED indicates 'Activity' by blinking high when the port receives or transmits a frame. P10_CRS is internally pulled low via resistor so the pin can be left unconnected when it is not used. The direction of this pin is determined by the value of the P10_MODE[2:0] pins at the rising edge of RESETn (during RESETn this pin is an input).
128	153	P10_COL /P10_SDET	Input	Collision. Collision is used to indicate both transmit and receive are occurring at the same time in half-duplex mode. The Collision signal is not synchronous to RXCLK. The Collision signal is used for half-duplex modes only and is ignored when the port is in full-duplex. When Port 10 is configured for 1000BASE-X operation (see P10_MODE[2:0]) Px_SDET is an input used to indicate whether a signal is detected by the fiber-optic transceiver. A positive level indicates that a signal is detected. P10_COL is internally pulled low via resistor so the pin can be left unconnected when it is not used.

Table 10: GMII/MII Transmit Interface (Port 10)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
111	135	P10_GTXCLK /MGMII	Output	Transmit Clock. GTXCLK is a reference for TXEN and TXD[7:0] when the port is in GMII mode. The speed of GTXCLK is 125 MHz and is normally only driven when the speed of the port is 1000BASE. GTXCLK can be configured to output a 25 MHz or 2.5 MHz clock so it can be used as a clock source for P10_TXCLK and P10_RXCLK when no other clock sources are available (see P10_MODE[2:0]). P10_GTXCLK is a multi-function pin used to configure the device during a hardware reset. When reset is asserted, this pin becomes an input and the configuration information below is latched at the rising edge of RESETn: 0 = All SERDES ports run in SGMII mode 1 = All SERDES ports run in MGMII mode - MGMII mode is recommended when connecting to Marvell® PHY devices that support auto-media detect when connecting to SERDES ports. MGMII mode uses out of band communication for link, speed, and flow control status, and works specifically with Marvell PHYs that support MGMII. P10_GTXCLK is tri-stated during RESETn and it is internally pulled high so the pin can be left unconnected if not used. **NOTE:** This pin can also be used as the RGMII Transmit Clock pin as documented in Table 12.
112	136	P10_TXCLK	Input	Transmit Clock. TXCLK is a reference for TXEN and TXD[3:0] when the port is in MII mode. The speed of TXCLK is 25 MHz or 2.5 MHz depending upon the speed of the port. P10_TXCLK is tri-stated during RESETn and it is internally pulled high so the pin can be left unconnected if not used.

Table 10: GMII/MII Transmit Interface (Port 10) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
101	125	P10_TXD[7] /ADDR[4]	Typically Output	P10_TXD[7:0]/P10_MODE[2:0]/ADDR[4:0] Transmit Data. TXD[7:0] outputs the data octet to be transmitted from the switch. TXD[7:0] is synchronous to GTXCLK in 1000BASE mode. In 100BASE and 10BASE modes, TXD[3:0] is synchronous to TXCLK and TXD[7:4] is ignored. P10_TXD[7:0] are multi-function pins used to configure the device during a hardware reset. When reset is asserted, these pins become inputs and the configuration information is latched at the rising edge of RESETn: TXD[2:0] = P10_MODE[2:0] TXD[7:3] = ADDR[4:0] P10_MODE[2:0] sets Port 10's mode of operation as follows: 000 = GMII with P10_GTXCLK = 125 MHz (1000BASE) 001 = RGMII = P10_TXC = 125 MHz (See Table 11 and Table 12 for RGMII pin descriptions. 010 = MII with P10_GTXCLK = 25 MHz (100BASE PHY Mode) 011 = MII with P10_GTXCLK = 2.5 MHz (10BASE PHY Mode) 100 = Cross-chip SERDES port with all GMII/MII pins tri-stated 101 = 1000BASE-X with all GMII/MII pins tri-stated (except for P10_LED) 110 = SGMII or MGMII with all GMII/MII pins tri-stated - See "P10_GTXCLK/MGMII" pin description in Table 10, "GMII/MII Transmit Interface (Port 10)," on page 39 for MGMII mode details. 111 = Port 10 is disabled (with all GMII/MII pins tri-stated) ADDR[4:0] sets the device's SMI address. If ADDR[4:0] are all 0's the device is configured in single device addressing mode. P10_TXD[7:0] are internally pulled low via resistor so the pins can be left unconnected when they are not used. Use a 4.7 kΩ resistor to VDDO for a configuration high. **NOTE:** P10_TXD[3:0] can also be used as the RGMII Transmit Data pins as documented in Table 12.
102	126	P10_TXD[6] /ADDR[3]		
103	127	P10_TXD[5] /ADDR[2]		
104	128	P10_TXD[4] /ADDR[1]		
106	130	P10_TXD[3] /ADDR[0]		
107	131	P10_TXD[2] /P10_MODE[2]		
108	132	P10_TXD[1] /P10_MODE[1]		
109	133	P10_TXD[0] /P10_MODE[0]		

Table 10: GMII/MII Transmit Interface (Port 10) (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
110	134	P10_TXEN/ P10_HALFDPX	Typically Output	Transmit Enable. Transmit enable is used to indicate when TXD[7:0] (or TXD[3:0] where appropriate) contains frame information. TXEN is synchronous to GTXCLK in 1000BASE mode and its synchronous to TXCLK in 100BASE and 10BASE modes. P10_TXEN is a multi-function pin used to configure the device during a hardware reset. When reset is asserted, this pin becomes an input and the configuration information is latched at the rising edge of RESETn: 0 = Sets Port 10 in full-duplex operation 1 = Sets Port 10 in half-duplex operation only if P10_MODE[2:0] = 01X P10_TXEN is internally pulled low via resistor so the pin can be left unconnected to select full-duplex on Port 10 or when it is not used. **NOTE:** This pin can also be used as the RGMII Transmit Control pin as documented in Table 12.

Port 10's RGMII receive interface supports 1000BASE-T, full-duplex mode of operation only. Port 10's RGMII receive interface uses the same pins as Port 10's GMII receive interface.

Table 11: RGMII Receive Interface (Port 10 Only)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
114	138	P10_RXC	Input	RGMII Receive Clock. In RGMII mode, P10_RXC provides a 125 MHz reference clock. P10_RXC is internally pulled high so the pin can be left unconnected if not used. **NOTE:** This pin can also be used as the GMII/MII Receive Clock pin as documented in Table 9.
120 119 118 117	144 143 142 141	P10_RD[3] P10_RD[2] P10_RD[1] P10_RD[0]	Input	RGMII Receive Data. P10_RD[3:0] run at double data rate with bits [3:0] presented on the rising edge of P10_RXC, and bits [7:4] presented on the falling edge of P10_RXC. P10_RD[3:0] are internally pulled low via resistor so the pins can be left unconnected when they are not used. **NOTE:** These pins can also be used as the GMII/MII Receive Data pins as documented in Table 9.
116	140	P10_RX_CTL	Input	RGMII Receive Control. In RGMII mode, P10_RX_CTL is presented on the rising edge of P10_RXC, and a logical derivative of P10_RX_DV and P10_RX_ER are presented on the falling edge of P10_RXC. P10_RX_CTL is internally pulled low via resistor so the pin can be left unconnected when it is not used. **NOTE:** This pin can also be used as the GMII/MII Receive Data Valid pin as documented in Table 9.

Port 10's RGMII transmit interface supports 1000BASE-T, full-duplex mode of operation only. Port 10's RGMII transmit interface uses the same pins as Port 10's GMII transmit interface.

Table 12: RGMII Transmit Interface (Port 10 Only)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
111	135	P10_TXC	Output	RGMII Transmit Clock. P10_TXC provides a 125 MHz reference clock. P10_TXC is tri-stated during RESETn and it is internally pulled high so the pin can be left unconnected if not used. **NOTE:** This pin can also be used as the GMII/MII Transmit Clock pin as documented in Table 10.
106 107 108 109	130 131 132 133	P10_TD[3] P10_TD[2] P10_TD[1] P10_TD[0]	Typically Output	RGMII Transmit Data. P10_TD[3:0] run at double data rate with bits [3:0] presented on the rising edge of P10_TXC, and bits [7:4] presented on the falling edge of P10_TXC. **NOTE:** These pins can also be used as the GMII/MII Transmit Data pins as documented in Table 10.
110	134	P10_TX_CTL	Typically Output	RGMII Transmit Control. In RGMII mode, P10_TX_CTL is presented on the rising edge of P10_TXC, and a logical derivative of P10_TX_EN and P10_TX_ER are presented on the falling edge of P10_TXC. **NOTE:** This pin can also be used as the GMII/MII Transmit Enable pin as documented in Table 10.

Table 13: Regulator & Reference

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
174	212	IREF	Analog	Current reference. A 2kΩ 1% resistor is placed between the IREF and VSS. This resistor is used to set an internal bias reference current.
67	77	CONTROL_12	Analog	Voltage control to external 1.2V regulator. This signal controls an external PNP transistor to generate the 1.2V power supply (from the 2.5V power rail) for the VDD_CORE pins. This pin is connected to the base of the PNP transistor, while the PNP transistor's collector will be connected to 1.2V and the emitter is connected to 2.5V
65	75	CONTROL_15	Analog	Voltage control to external 1.5V regulator. This signal controls an external PNP transistor to generate the 1.5V power supply (from the 2.5V power rail). This pin is connected to the base of the PNP transistor, while the PNP transistor's collector will be connected to 1.5V and the emitter is connected to 2.5V If this pin is not used, it must be left unconnected.
66	76	FEEDBACK_15	Analog	Voltage feedback for 1.5V regulator. This signal must be connected to the output of the external PNP transistor that is connected to the CONTROL_15 pin, if the CONTROL_15 pin is used. If the CONTROL_15 pin is not used, this pin must be left unconnected.

Table 14: System

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
70	80	XTAL_IN	Input	25 MHz system reference clock input provided from the board. The clock source can come from an external crystal or an external oscillator. This is the only clock required. Refer to Section 16.7.2 for XTAL_IN timing requirements.
69	79	XTAL_OUT	Output	System reference clock output provided to the board. This output can only be used to drive an external crystal. It cannot be used to drive external logic. If an external oscillator is used this pin should be left unconnected.
97	121	INTn	Open Drain Output	INTn is an active low, open drain pin that is asserted to indicate an unmasked interrupt event occurred. A single external pull-up resistor is required somewhere on this interrupt net for it to go high when it is inactive. The INTn pin will go active low if SW_MODE[1:0] (Table 17) is not 0b10 (stand alone mode) and if the EEPROM data has executed a Halt OpCode.
96	120	RESETn	Input	Hardware reset. Active low. The device is configured during reset. When RESETn is low, all configuration pins become inputs and the value seen on these pins is latched on the rising edge of RESETn or some time after. Refer to Section 16.7.1 for Reset and Configuration Timing details.

Table 15: Register Access Interface

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
94	118	MDC_PHY /LRN2ALL	Typically Output	Management Data Clock, Master. MDC_PHY is the reference clock output for the serial management interface (SMI) that connects to an external SMI slave device, typically external PHYs. The Master SMI is used to access the registers in the external PHYs and it is available to the CPU to use (Global 2, offsets 0x18 and 0x19). MDC_PHY is a multi-function pin used to configure the device during a hardware reset. When reset is asserted, MDC_PHY becomes an input and configuration information is latched into the device. If the SW_24P configuration pin (Table 8) is low at the rising edge of RESETn, this pin must be high. If the SW_24P configuration pin (Table 8) is high at the rising edge of RESETn the ATU's LRN2ALL bit is configures as follows: Low = Do not send learning update frames to Distributed Switching Architecture (DSA) Tag ports High = Send learning update frames to DSA Tag ports MDC_PHY is internally pulled high via a resistor so it can be left floating when unused. **NOTE:** On previous pin compatible devices, this pin functioned as a PPU_EN configuration pin. While this pin still performs this function, it is now recommended that the PPU always remain enabled and then access the PHY registers using the SMI PHY Command and Data registers (Global 2, offsets 0x18 and 0x19). The PPUState bits (Global 1, offset 0x00) and the PPUEn bit (Global 1, offset 0x04) exist in this device but are marked as Reserved, or they are modified, as these bits are not expected to be used as they were in the previous devices. The PPUEn and PPUState bits will physically be changed in future devices to match this documentation.

Table 15: Register Access Interface (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
95	119	MDIO_PHY	I/O	Management Data I/O, Master. MDIO_PHY is used to transfer management data in and out of the device synchronously to MDC_PHY. This pin requires an external pull-up resistor in the range of 4.7 kΩ to 10 kΩ. The device uses Device Addresses 0x08, 0x09, and 0x0A to access the external PHYs for ports 8, 9, and 10 respectively. The CPU can read or write any register on any device connected to this MDC_PHY/MDIO_PHY interface - see the SMI PHY Command and Data registers (Global 2, offsets 0x18 and 0x19) MDIO_PHY is internally pulled high via a resistor so it can be left floating when unused.
99	123	MDC_CPU	Input	Management Data Clock, Slave. MDC_CPU is the reference clock input for the serial management interface (SMI) that connects to an external SMI master, typically a CPU. A continuous clock stream is not expected. The maximum frequency supported is 8.3 MHz. The CPU's SMI interface is used to access the registers in the switch and it is available in all combinations of SW_MODE[1:0] (Table 17). MDC_CPU is internally pulled high via a resistor so it can be left floating when unused.
98	122	MDIO_CPU	I/O	Management Data I/O, Slave. MDIO_CPU is used to transfer management data in and out of the device synchronously to MDC_CPU. This pin requires an external pull-up resistor in the range of 4.7 kΩ to 10 kΩ. The device uses one or all of the 32 possible SMI port addresses (two modes are supported). The address(es) that are used are selectable using the P10_TXD[4:0]/ADDR[4:0] pins. MDIO_CPU is internally pulled high via a resistor so it can be left floating when unused.

Table 16: Serial EEPROM Interface

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
138	171	EE_CS /EE_4WIRE	Typically Output	Serial EEPROM chip select. EE_CS is the serial EEPROM chip select referenced to EE_CLK. It is used to enable the external EEPROM (if present), and to delineate each data transfer. EE_CS is a multi-function pin used to configure the device during a hardware reset. When reset is asserted, EE_CS becomes an input and the desired EEPROM type configuration is latched at the rising edge of RESETn as follows: Low = Supports 2-Wire EEPROMs (For 1K bit 24C01A, 2K bit 24C02, & 4K bit 24C04 High = Supports 4-Wire EEPROMs (For 2K bit 93C56 & 4K bit 93C66 The external EEPROM must be configured in the x16 organization using 8-bit addresses. EE_CS is internally pulled high via a resistor so the pin can be left floating when unused or to select support for 4-Wire EEPROMs. Use a 4.7kΩ resistor to VSS for a configuration low to support 2-Wire EEPROMs.

Warning

When 2-Wire EEPROM mode is selected, a 2-Wire EEPROM device must be connected to the device or the device will not reset properly.

Table 16: Serial EEPROM Interface (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
137	170	EE_CLK /FD_FLOW_DIS	Typically Output	Serial EEPROM clock. EE_CLK is the serial EEPROM clock reference output by the device. It is used to shift the external serial EEPROM (if installed) to the next data bit so the default values of the internal registers can be overridden. EE_CLK is a multi-function pin used to configure the device during a hardware reset. When reset is asserted, EE_CLK becomes an input and it becomes the full-duplex Flow Control disable as follows: Low = Enable advertisement of full-duplex flow control on all PHYs High = Disable full-duplex flow control on all full-duplex ports Full-duplex flow control requires support from the end station. It is supported on any full-duplex port that has Auto-Negotiation enabled, advertises that it supports Pause (i.e., FD_FLOW_DIS = Low at reset), and sees that the end station also supports Pause (from data returned during Auto-Negotiation). EE_CLK is internally pulled high via a resistor so the pin can be left unconnected for a configuration high. Use a 4.7 kΩ resistor to VSS for a configuration low.

Table 16: Serial EEPROM Interface (Continued)

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
136	169	EE_DIN /HD_FLOW_DIS	Typically Input Output During EEPROM Loading	Serial EEPROM data into the EEPROM device. EE_DIN is serial EEPROM data referenced to EE_CLK used to transmit the EEPROM command and address to the external serial EEPROM (if present). After the serial EEPROM data is loaded EE_DIN becomes half-duplex Flow Control disable as follows: Low = Enable “forced collision” flow control on all half-duplex ports High = Disable flow control on all half-duplex ports Half-duplex flow control is active on all half-duplex ports whenever this pin is low. HD_FLOW_DIS is latched after reset. EE_DIN is internally pulled high via a resistor so it can be left floating to disable half-duplex flow control or when the pin is unused. Use a 4.7 kΩ resistor to VSS for a configuration low.
133	166	EE_DOUT	Input	Serial EEPROM data out from the EEPROM device. EE_DOUT is serial EEPROM data referenced to EE_CLK used to receive the EEPROM configuration data from the external serial EEPROM (if present). EE_DOUT is internally pulled high via a resistor so it can be left floating when the pin is unused.

Table 17: Switch Configuration Interface

88E6096/ 88E6097	88E6097F	Pin Name	Type	Description
131 132	156 157	SW_MODE[1] SW_MODE[0]	Input	Switch Mode. These pins are used to configure the device after reset. Switch Mode pins work as follows: 1 0 Description 0 0 CPU attached mode – ports come up disabled[1] 0 1 Reserved 1 0 Test mode – ports come up enabled – ignore EEPROM 1 1 EEPROM attached mode – EEPROM defined port states[2] The EEPROM attached mode (when both SW_MODE pins = high) can be used with a CPU and the CPU attached mode can be used with an EEPROM. In these modes, the INTn pin will go active low after the EEPROM is done initializing the internal registers. SW_MODE[1:0] are not latched on the rising edge of RESETn and they must remain static for proper device operation. They are internally pulled high via resistors so the pins can be left unconnected to enable the EEPROM attached mode.

1. The ports come up disabled in the CPU mode so the CPU can perform bridge loop detection on link up.
2. In EEPROM attached mode, the ports come up enabled unless the Port Control register is overwritten by the EEPROM data (see Section 14 for EEPROM programming format details).

Table 18: Power & Ground

88E6096/ 88E6097	88E6097F	PIN NAME	Type	Description
91	115	VDDO_PHY	Power	Power to the MDC_PHY and MDIO_PHY outputs (pin numbers: 91–95 in the 88E6096/88E6097 devices; 115–119 in the 88E6097F device). VDDO_PHY can be connected to 3.3V to support 3.3V I/O or 2.5V for 2.5V I/O.
72 80	86 95 104 106	VDDO_P9	Power	Power to Port 9's GMII/MII outputs (pin numbers: 71–90 in the 88E6096/88E6097 devices; 81–114 in the 88E6097F device). VDDO_P9 can be connected to 3.3V to support 3.3V I/O or 2.5V for 2.5V I/O.
105 113 121	129 137 145 146	VDDO_P10	Power	Power to Port 10's GMII/MII outputs (pin numbers: 96–128 in the 88E6096/88E6097 devices; 120–153 in the 88E6097F device). VDDO_P10 can be connected to 3.3V to support 3.3V I/O or 2.5V for 2.5V I/O.
135	168	VDDO_ EEPROM	Power	Power to the serial EEPROM outputs (pin numbers: 129-139 in the 88E6096/88E6097 devices; 154-172 in the 88E6097F device). VDDO_EEPROM can be connected to 3.3V to support 3.3V I/O or 2.5V for 2.5V I/O.
145 154 162 170	178 187 195 203	VDDO_LED	Power	Power to LED outputs (pin numbers: 140-173 in the 88E6096/88E6097 devices; 173-206 in the 88E6097F device) VDDO_LED can be connected to 3.3V to support 3.3V I/O or 2.5V for 2.5V I/O.
	54	VDDO_SLED	Power	Power to serial LED outputs (pin numbers: 51–54 in the 88E6097Fdevice). VDDO_SLED can be connected to 3.3V to support 3.3V I/O or 2.5V for 2.5V I/O.
68	78	VDD_PLL	Power	2.5V Power to the internal PLL blocks inside the device.
48 54 60	58 64 70	P8_VDDAH P9_VDDAH P10_VDDAH	Power	2.5V Power to analog core used to power the SERDES interface for Port 8, Port 9 and Port 10. If the port's SERDES is not used, Px_VDDAH can be tied to VSS to save power. If Port 9 or Port 10 is configured as a digital interface, P9_VDDAH and/or P10_VDDAH must be connected to 2.5V, respectively.

Table 18: Power & Ground (Continued)

88E6096/ 88E6097	88E6097F	PIN NAME	Type	Description
49 55 61	59 65 71	P8_VTT P9_VTT P10_VTT	Power	1.2V or 1.5V Power to analog core used to power each SERDES transmitter. Refer to Table 187 or Table 188 - "Transmitter DC Characteristics" for SERDES output drive levels and how they are affected by Px_VTT. Use 1.2V power for chip-to-chip connections. Use 1.5V power for 1000BASE-X or backplane applications. For lowest power, P8_VTT, P9_VTT and/or P10_VTT may be left floating, respectively.
11 22 33 44 176	1 12 23 34 46 211 216	VDDAH	Power	2.5V Power to the analog core used to power each 10/100 PHY interface.
3 8 14 19 25 30 36 41	4 9 15 20 26 31 37 42	VDDAL	Power	1.2V Power to analog core used to power each 10/100 PHY interface.
45 64 71 78 85 100 115 134 139 146 151 158 168	45 55 74 81 92 100 105 124 139 167 172 179 184 191 201	VDD_CORE	Power	1.2V Power to digital core

Table 18: Power & Ground (Continued)

88E6096/ 88E6097	88E6097F	PIN NAME	Type	Description
EPAD	EPAD	VSS	Ground	Ground to device. The 88E6096/88E6097 devices are contained in the 176 TQFP package, while the 88E6097F device is contained in a 216 LQFP package. Both the 176 TQFP and 216 LQFP packages have an exposed die pad (E-PAD) at their base. The EPAD must be soldered to VSS as it is the only VSS connection on the device. The location and dimensions of the EPAD can be found in Figure 104 and Table 215 for the 176 TQFP, and Figure 105 and Table 216 for the 216 LQFP, respectively.
129 130 175	107 108 154 155 213 214 215	NC		No Connect. Do not connect these pins to anything.

1.2 88E6096 and 88E6097 Device Pin Assignment List

Table 19: Package Pin List—Alphabetical by Signal Name

Pin Number	Pin Name
140	CONFIG_A
141	CONFIG_B
67	CONTROL_12
65	CONTROL_15
137	EE_CLK/FD_FLOW_DIS
138	EE_CS/EE_4WIRE
136	EE_DIN/HD_FLOW_DIS
133	EE_DOUT
66	FEEDBACK_15
97	INTn
174	IREF
99	MDC_CPU
94	MDC_PHY/LRN2ALL
98	MDIO_CPU
95	MDIO_PHY
129	NC
130	NC
175	NC
142	P0_LED0
143	P0_LED1
144	P0_LED2
2	P0_RXN
1	P0_RXP
5	P0_TXN
4	P0_TXP
147	P1_LED0
148	P1_LED1
149	P1_LED2
9	P1_RXN
10	P1_RXP
6	P1_TXN
7	P1_TXP
150	P2_LED0
152	P2_LED1
153	P2_LED2
13	P2_RXN
12	P2_RXP
16	P2_TXN
15	P2_TXP
155	P3_LED0
156	P3_LED1
157	P3_LED2
20	P3_RXN
21	P3_RXP
17	P3_TXN
18	P3_TXP
159	P4_LED0
160	P4_LED1
161	P4_LED2
24	P4_RXN
23	P4_RXP
27	P4_TXN
26	P4_TXP
163	P5_LED0
164	P5_LED1
165	P5_LED2
31	P5_RXN
32	P5_RXP

Pin Number	Pin Name
28	P5_TXN
29	P5_TXP
166	P6_LED0
167	P6_LED1
169	P6_LED2
35	P6_RXN
34	P6_RXP
38	P6_TXN
37	P6_TXP
171	P7_LED0
172	P7_LED1
173	P7_LED2
42	P7_RXN
43	P7_RXP
39	P7_TXN
40	P7_TXP
92	P8_LED/P8_MODE
46	P8_RXN
47	P8_RXP
93	P8_SDET
50	P8_TXN
51	P8_TXP
48	P8_VDDAH
49	P8_VTT
90	P9_COL/P9_SDET
89	P9_CRS/P9_LED
81	P9_RXCLK
83	P9_RXD[0]
84	P9_RXD[1]
86	P9_RXD[2]

Pin Number	Pin Name
87	P9_RXD[3]
82	P9_RXDV
88	P9_RXER
52	P9_RXN
53	P9_RXP
79	P9_TXCLK
76	P9_TXD[0]/P9_MODE[0]
75	P9_TXD[1]/P9_MODE[1]
74	P9_TXD[2]/P9_MODE[2]
73	P9_TXD[3]/SW_24P
77	P9_TXEN/P9_HALFDPX
56	P9_TXN
57	P9_TXP
54	P9_VDDAH
55	P9_VTT
128	P10_COL/P10_SDET
127	P10_CRS/P10_LED
111	P10_GTXCLK/MGMII/P10_TXC
114	P10_RXCLK/P10_RXC
117	P10_RXD[0]/P10_RD[0]
118	P10_RXD[1]/P10_RD[1]
119	P10_RXD[2]/P10_RD[2]
120	P10_RXD[3]/P10_RD[3]
122	P10_RXD[4]
123	P10_RXD[5]
124	P10_RXD[6]
125	P10_RXD[7]
116	P10_RXDV/P10_RX_CTL
126	P10_RXER
58	P10_RXN

Pin Number	Pin Name
59	P10_RXP
112	P10_TXCLK
109	P10_TXD[0]/P10_MODE[0]/ P10_TD[0]
108	P10_TXD[1]/P10_MODE[1]/ P10_TD[1]
107	P10_TXD[2]/P10_MODE[2]/ P10_TD[2]
106	P10_TXD[3]/ADDR[0]/P10_TD[3]
104	P10_TXD[4]/ADDR[1]
103	P10_TXD[5]/ADDR[2]
102	P10_TXD[6]/ADDR[3]
101	P10_TXD[7]/ADDR[4]
110	P10_TXEN/P10_HALFDPX/ P10_TX_CTL
62	P10_TXN
63	P10_TXP
60	P10_VDDAH
61	P10_VTT
96	RESETn
132	SW_MODE[0]
131	SW_MODE[1]
45	VDD_CORE
64	VDD_CORE
71	VDD_CORE
78	VDD_CORE
85	VDD_CORE
100	VDD_CORE
115	VDD_CORE
134	VDD_CORE
139	VDD_CORE
146	VDD_CORE

Pin Number	Pin Name
151	VDD_CORE
158	VDD_CORE
168	VDD_CORE
68	VDD_PLL
11	VDDAH
22	VDDAH
33	VDDAH
44	VDDAH
176	VDDAH
3	VDDAL
8	VDDAL
14	VDDAL
19	VDDAL
25	VDDAL
30	VDDAL
36	VDDAL
41	VDDAL
135	VDDO_EEPROM
145	VDDO_LED
154	VDDO_LED
162	VDDO_LED
170	VDDO_LED
72	VDDO_P9
80	VDDO_P9
105	VDDO_P10
113	VDDO_P10
121	VDDO_P10
91	VDDO_PHY
EPAD	VSS
70	XTAL_IN
69	XTAL_OUT

1.3 88E6097F Device Pin Assignment List

Table 20: Package Pin List—Alphabetical by Signal Name

Pin Number	Pin Name
173	CONFIG_A
174	CONFIG_B
77	CONTROL_12
75	CONTROL_15
170	EE_CLK/FD_FLOW_DIS
171	EE_CS/EE_4WIRE
169	EE_DIN/HD_FLOW_DIS
166	EE_DOUT
76	FEEDBACK_15
121	INTn
212	IREF
53	LEDCLK
51	LEDENA
52	LEDSER
123	MDC_CPU
118	MDC_PHY/LRN2ALL
122	MDIO_CPU
119	MDIO_PHY
107	NC
108	NC
154	NC
155	NC
213	NC
214	NC
215	NC
165	P0_CONFIG

Pin Number	Pin Name
175	P0_LED0
176	P0_LED1
177	P0_LED2
3	P0_RXN
2	P0_RXP
207	P0_SDET
6	P0_TXN
5	P0_TXP
164	P1_CONFIG
180	P1_LED0
181	P1_LED1
182	P1_LED2
10	P1_RXN
11	P1_RXP
208	P1_SDET
7	P1_TXN
8	P1_TXP
163	P2_CONFIG
183	P2_LED0
185	P2_LED1
186	P2_LED2
14	P2_RXN
13	P2_RXP
209	P2_SDET
17	P2_TXN
16	P2_TXP

Pin Number	Pin Name
162	P3_CONFIG
188	P3_LED0
189	P3_LED1
190	P3_LED2
21	P3_RXN
22	P3_RXP
210	P3_SDET
18	P3_TXN
19	P3_TXP
161	P4_CONFIG
192	P4_LED0
193	P4_LED1
194	P4_LED2
25	P4_RXN
24	P4_RXP
47	P4_SDET
28	P4_TXN
27	P4_TXP
160	P5_CONFIG
196	P5_LED0
197	P5_LED1
198	P5_LED2
32	P5_RXN
33	P5_RXP
48	P5_SDET
29	P5_TXN
30	P5_TXP
159	P6_CONFIG

Pin Number	Pin Name
199	P6_LED0
200	P6_LED1
202	P6_LED2
36	P6_RXN
35	P6_RXP
49	P6_SDET
39	P6_TXN
38	P6_TXP
158	P7_CONFIG
204	P7_LED0
205	P7_LED1
206	P7_LED2
43	P7_RXN
44	P7_RXP
50	P7_SDET
40	P7_TXN
41	P7_TXP
116	P8_LED/P8_MODE
56	P8_RXN
57	P8_RXP
117	P8_SDET
60	P8_TXN
61	P8_TXP
58	P8_VDDAH
59	P8_VTT
110	P9_COL/P9_SDET
109	P9_CRS/P9_LED
93	P9_GTXCLK

Pin Number	Pin Name
96	P9_RXCLK
98	P9_RXD[0]
99	P9_RXD[1]
101	P9_RXD[2]
102	P9_RXD[3]
111	P9_RXD[4]
112	P9_RXD[5]
113	P9_RXD[6]
114	P9_RXD[7]
97	P9_RXDV
103	P9_RXER
62	P9_RXN
63	P9_RXP
94	P9_TXCLK
90	P9_TXD[0]/P9_MODE[0]
89	P9_TXD[1]/P9_MODE[1]
88	P9_TXD[2]/P9_MODE[2]
87	P9_TXD[3]/SW_24P
85	P9_TXD[4]
84	P9_TXD[5]
83	P9_TXD[6]
82	P9_TXD[7]
91	P9_TXEN/P9_HALFDPX
66	P9_TXN
67	P9_TXP
64	P9_VDDAH
65	P9_VTT
153	P10_COL/P10_SDET

Pin Number	Pin Name
152	P10_CRS/P10_LED
135	P10_GTXCLK/MGMII/P10_TXC
138	P10_RXCLK/P10_RXC
141	P10_RXD[0]/P10_RD[0]
142	P10_RXD[1]/P10_RD[1]
143	P10_RXD[2]/P10_RD[2]
144	P10_RXD[3]/P10_RD[3]
147	P10_RXD[4]
148	P10_RXD[5]
149	P10_RXD[6]
150	P10_RXD[7]
140	P10_RXDV/P10_RX_CTL
151	P10_RXER
68	P10_RXN
69	P10_RXP
136	P10_TXCLK
133	P10_TXD[0]/P10_MODE[0]/ P10_TD[0]
132	P10_TXD[1]/P10_MODE[1]/ P10_TD[1]
131	P10_TXD[2]/P10_MODE[2]/ P10_TD[2]
130	P10_TXD[3]/ADDR[0]/P10_TD[3]
128	P10_TXD[4]/ADDR[1]
127	P10_TXD[5]/ADDR[2]
126	P10_TXD[6]/ADDR[3]
125	P10_TXD[7]/ADDR[4]
134	P10_TXEN/P10_HALFDPX/ P10_TX_CTL
72	P10_TXN

Pin Number	Pin Name
73	P10_TXP
70	P10_VDDAH
71	P10_VTT
120	RESETn
157	SW_MODE[0]
156	SW_MODE[1]
45	VDD_CORE
55	VDD_CORE
74	VDD_CORE
81	VDD_CORE
92	VDD_CORE
100	VDD_CORE
105	VDD_CORE
124	VDD_CORE
139	VDD_CORE
167	VDD_CORE
172	VDD_CORE
179	VDD_CORE
184	VDD_CORE
191	VDD_CORE
201	VDD_CORE
78	VDD_PLL
1	VDDAH
12	VDDAH
23	VDDAH
34	VDDAH
46	VDDAH
211	VDDAH

Pin Number	Pin Name
216	VDDAH
4	VDDAL
9	VDDAL
15	VDDAL
20	VDDAL
26	VDDAL
31	VDDAL
37	VDDAL
42	VDDAL
168	VDDO_EEPROM
178	VDDO_LED
187	VDDO_LED
195	VDDO_LED
203	VDDO_LED
86	VDDO_P9
95	VDDO_P9
104	VDDO_P9
106	VDDO_P9
129	VDDO_P10
137	VDDO_P10
145	VDDO_P10
146	VDDO_P10
115	VDDO_PHY
54	VDDO_SLED
EPAD	VSS
80	XTAL_IN
79	XTAL_OUT

Section 2. Application Examples

2.1 Examples using the 88E6097 Device

Figure 3: 16 FE + 2 GE Ports

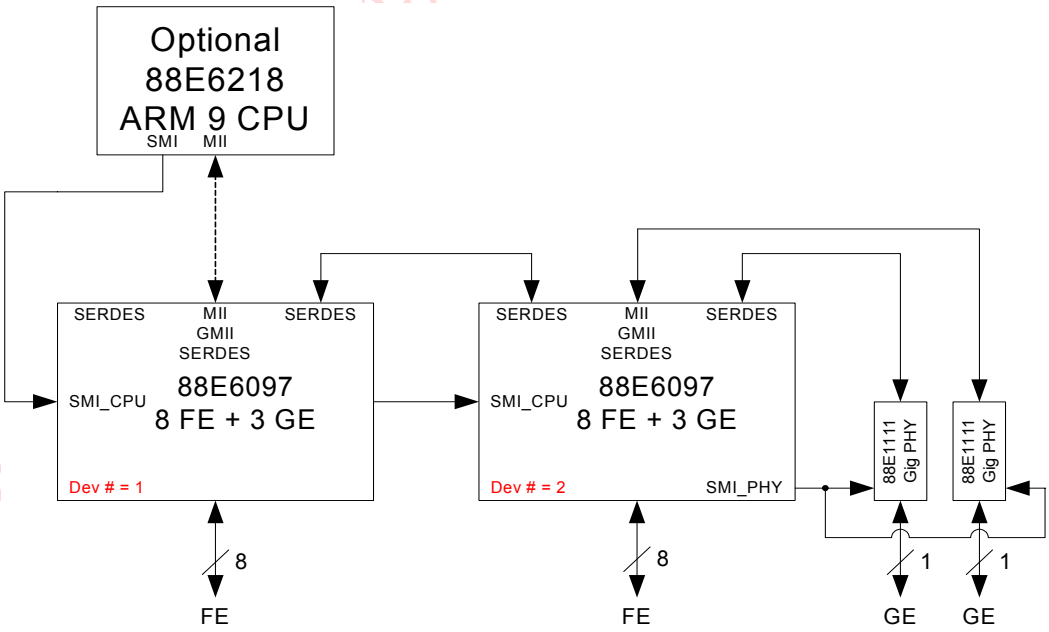


Figure 4: 24 FE + 2 GE with 88E6097 X 3

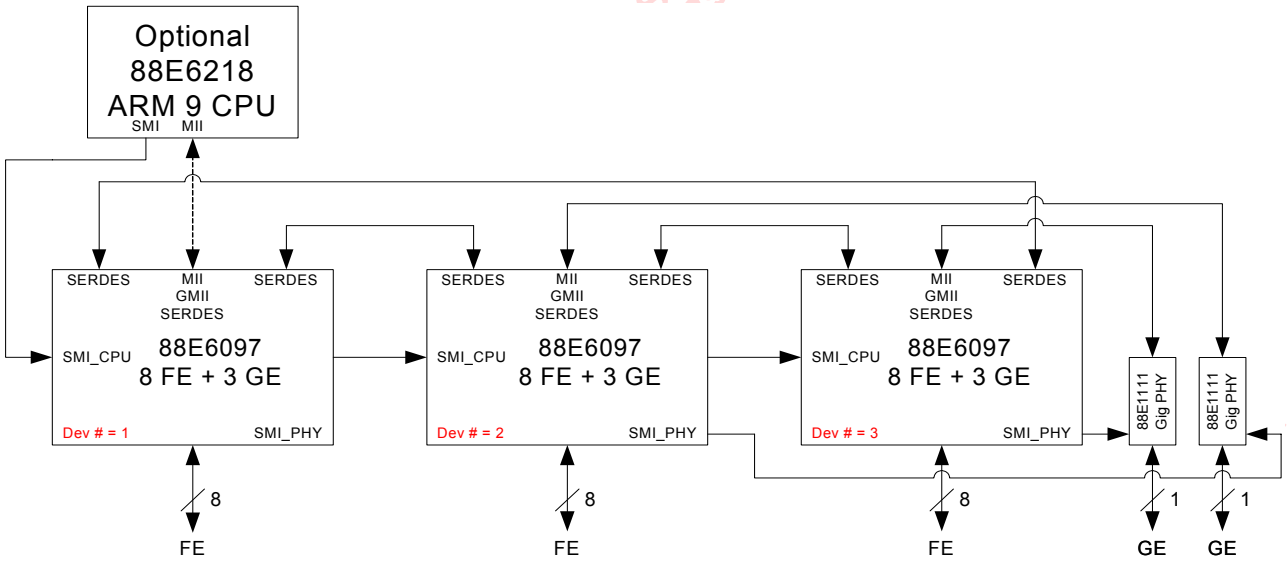


Figure 5: 48 FE + 4 GE wit 88E6097 X 6 + 88E6185

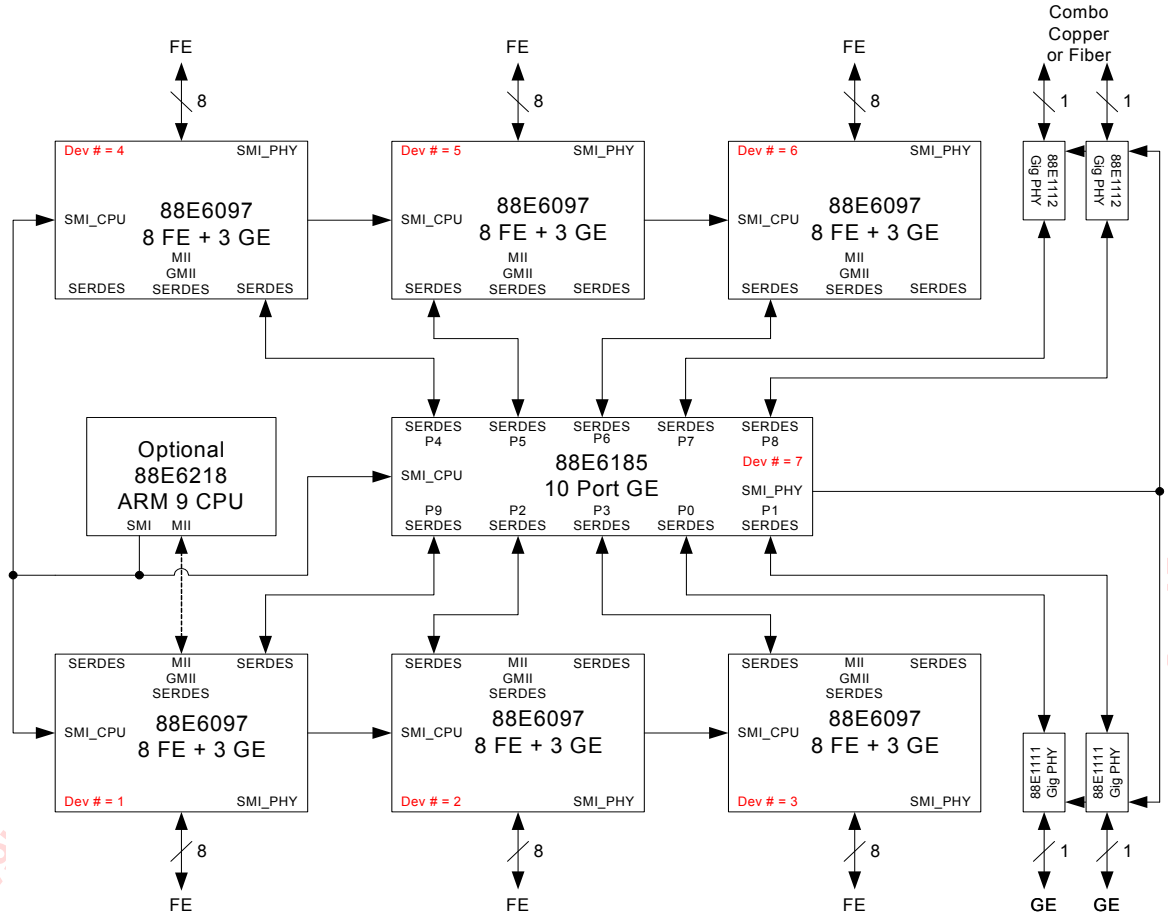


Figure 6: Fiber to the Curb

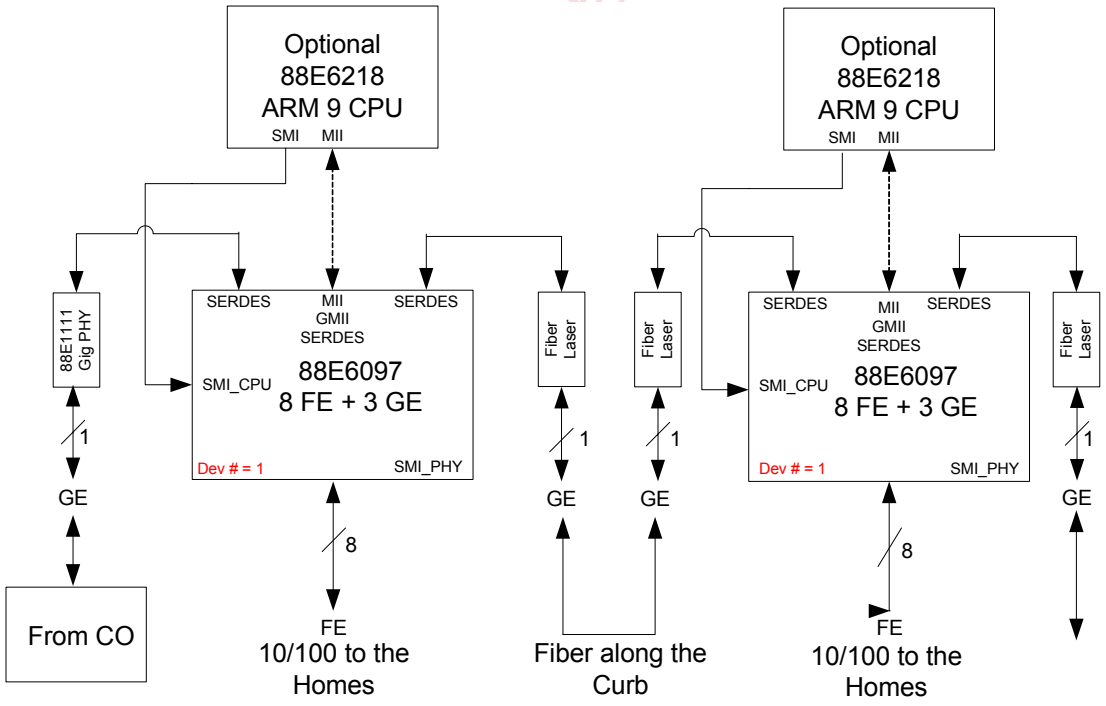
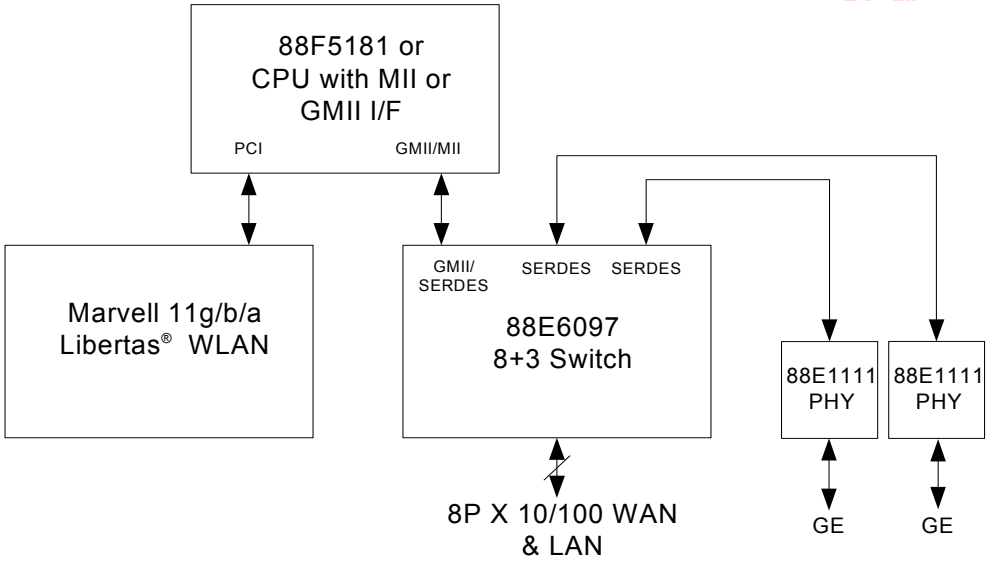


Figure 7: SOHO/SMB Router with FE and GE Ports



2.2 Device Physical Interfaces

The devices contain a number of interfaces that support both copper and fiber media. Table 21 lists the interfaces supported on each port of the 88E6096 and 88E6097 devices. Table 22 lists the interfaces supported on each port of the 88E6097F device. Refer to the diagrams further in this section for connection details.

Table 21: 88E6096/88E6097 Device Interfaces

Port	10BASE-T 100BASE-T[1]	1000BASE-T (w/ external PHY)	1000BASE-X SERDES	MAC or PHY Mode MII Only	MAC or PHY Mode GMII/MII	RGMII[2]
0-7	x					
8	x	x	x			
9	x	x	x	x		
10	x	x	x		x	x

1. Ports 8, 9, and 10 require an external PHY for these modes of operation.
2. 1000BASE-X, full-duplex only.

Table 22: 88E6097F Device Interfaces

Port	10BASE-T 100BASE-T[1]	1000BASE-T (w/ external PHY)	1000BASE-X SERDES	100BASE-FX	MAC or PHY Mode GMII/MII	RGMII[2]
0-7	x			x		
8	x	x	x			
9	x	x	x		x	
10	x	x	x		x	x

1. Ports 8, 9, and 10 require an external PHY for these modes of operation.
2. 1000BASE-X, full-duplex only.

2.2.1 10/100 PHY Interface

Ports 0 to 7 on the devices support a 10/100 PHY interface. In the 88E6096/88E6097 devices this interface supports 10BASE-T and 100BASE-TX copper IEEE standards. Ports 0 to 7 on the 88E6097F device support a 100BASE-FX fiber option as well. The MAC inside the switch works the same way regardless of the external interface being used. Each PHY's Link, Speed, Duplex and Flow Control information is directly communicated to the MAC it is attached to so the MAC tracks, or follows, the mode the PHY links up in. A detailed description of the PHY is covered in Section 11, and the PHY registers are covered in Section 15.

2.2.2 SERDES or (G)MII Interface

Ports 8 to 10 in the devices are SERDES interfaces. The SERDES interfaces can be used for these options

- Connection to Marvell® triple speed 10/100/1000 copper PHYs
- Connection to 1000BASE-X fiber modules
- Cross-chip connection to other Marvell switch devices (another 88E6097 device)

The tenth and eleventh ports (Port 9 and Port 10), on the devices also support optional short-distance, industry-standard digital interfaces, referred to generically as the port's (G)MII or digital interface. Many interface modes and timings are supported so that a large number of external device types can be used. The 88E6096/ 88E6097 devices support an MII interface on Port 9 and an (G)MII/RGMII interface on Port 10, while the 88E6097F device supports a (G)MII interface on Port 9 and a (G)MII/RGMII interface on Port 10. Refer to Section 2.2.3, "Digital Interface Options," on page 69 for details.

2.2.2.1 Triple Speed PHY SERDES Interface Option

Port 8, 9 and 10's SERDES can be configured to use a triple speed PHY interface to an external PHY. In this mode, the SERDES use the SGMII protocol. The in-band Link, Speed and Duplex signals in the SGMII protocol are ignored. The external PHY's Link, Speed, Duplex and Flow Control information must be transferred to the port's MAC so the MAC is in the correct mode. This can be done in software (if the port's PHYDetect bit is zero - Port offset 0x00) or it is done automatically by the PHY Polling Unit (PPU - Section 2.2.4).

The triple speed PHY interface can support Marvell PHYs with Auto-Media Detect™ for auto switching between copper and fiber. This can be supported in software or automatically in hardware by the PHY Polling Unit (PPU) after setting the port's MGMII bit to a one (in the port's Port Status Register - offset 0x00). If the port's MGMII bit is not set to a one, the PPU will support copper only and will not support Auto-Media Detect.

Port 8 is set to Triple Speed PHY mode if its Px_LED/Px_MODE pin is low at the rising edge of RESETn and if the port's PHYDetect bit is a 1 (in the port's Port Status Register - offset 0x00). PHYDetect will be set to a 1 on Port 8 if the PPU finds a PHY at SMI address 0x08.

Ports 9 and 10 are set to Triple Speed PHY mode if their Px_TXD[2:0]/Px_MODE[2:0] pins are set to 0x6 at the rising edge of RESETn. The automatic transfer of PHY status to its MAC requires that the port's PHYDetect bit be a 1 (in the port's Port Status Register - offset 0x00). PHYDetect will be set to a 1 on Port 9 if the PPU finds a PHY at SMI address 0x09. It will be set to a 1 on Port 10 if the PPU finds a PHY at SMI address 0x0A.

2.2.2.2 IEEE 1000BASE-X SERDES Interface Option

Port 8, 9 and 10's SERDES can be configured in 1000BASE-X.

Port 8 is set to 1000BASE-X mode if its Px_LED/Px_MODE pin is high at the rising edge of RESETn. Connecting an LED to VDDO through a resistor to this pin will enable 1000BASE-X on the port. After RESETn the Px_LED/Px_MODE pin will become the Link/Activity LED for the port (off = no link, on = link, blink = activity).

Ports 9 and 10 are set to 1000BASE-X mode if their Px_TXD[2:0]/Px_MODE[2:0] pins are set to 0x5 at the rising edge of RESETn. In this mode, the Px_CRS/Px_LED pin become the Link/Activity LED for the port (off = no link, on = link, blink = activity) after RESETn.

The port enters 1000BASE-X mode, if configured, even if an external PHY is detected at the port's SMI address.

1000BASE-X mode uses a PCS to auto-negotiate with a link partner to determine if Flow Control should be supported or not (auto-negotiation can be disabled). Link will be automatically established if the port's Px_SDET is detected high and the port's PCS determines Sync is OK (sets the port's SyncOK to a 1). Link will automatically go down if either Px_SDET or SyncOK go to zero. Speed is always 1000 Mbps and Duplex is always full-duplex on 1000BASE-X ports. An interrupt can be generated on the ports when link changes state (see Global 2, offset 0x00 and 0x01).

2.2.2.3 Cross-chip Interface Option

Port 8, 9, and 10's SERDES can be configured in cross-chip mode. This mode is used to connect two or more Marvell® switch devices together to create a larger switch. 1000BASE-X protocol is used on the line and the port's Speed is locked to 1000 Mbps and its Duplex is fixed at full-duplex.

Port 8 is set to cross-chip mode if its Px_LED/Px_MODE pin is low at the rising edge of RESETn and if the port's PHYDetect bit is a 0 (in the port's Port Status Register - offset 0x00). PHYDetect will be set to a 0 on port 8 if the PPU cannot find a PHY at SMI address 0x08.

Port 9 and Port 10 are set to cross-chip mode if their Px_TXD[2:0]/Px_MODE[2:0] pins are set to 0x4 at the rising edge of reset.

Note

Ports that are configured in cross-chip mode are initialized with their Link down (unless the SW_24P configuration option is being used - Section 2.2.5). This gives software time to initialize the switch's configuration before software allows packets to flow by forcing the port's link up (using the port's ForcedLink bit in the port's PCS Control register - offset 0x01).

2.2.2.4 Port Status Registers

Each switch port of the devices has a status register that reports information about that port's MAC, SERDES or (G)MII interface. These registers can be used to check the current port configuration. See Table 71, Port Status Register, for more information.

2.2.3 Digital Interface Options

The (G)MII digital interface supports many different modes defined in the following sections. The mode to use is configured once at reset by external pull-up resistors connected to the P9_MODE[2:0] and P10_MODE[2:0] pins. If Port 9 or Port 10 is not connected to any device, the port should be disabled. If Port 9 or Port 10's (G)MII digital pins are unused but the port's SERDES interface is used, the P9_MODE[2:0] or P10_MODE[2:0] pins must be set accordingly. See Table 8 and Table 10 for more information.

For the 88E6096/88E6097 devices, Port 9 supports MII MAC and MII PHY Mode.

For the 88E6096/88E6097 and 88E6097F devices, Port 10 supports MII MAC mode, MII PHY mode, and (G)MII/RGMII mode options.

For the 88E6097F device, Port 9 supports MII MAC mode, MII PHY mode and (G)MII mode options.

Note

(G)MII PHY mode and (G)MII MAC mode are discussed in the following sections. Electrically, there is no difference since the GMII Interface uses source synchronous clocks. Each concept is discussed separately since the port supports being connected to an external PHY (GMII MAC mode - where the port looks like a MAC supporting 10/100/1000 Mbps) or to an external MAC (GMII PHY mode where the port looks like a PHY supporting 1000 Mbps only).

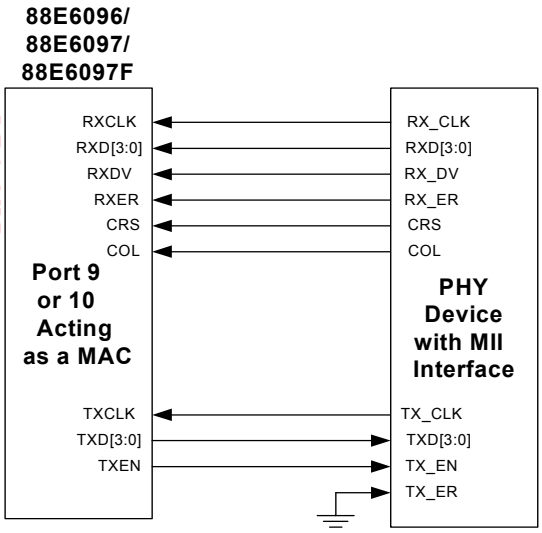
Warning

A port's SERDES must be powered up to generate the GTX_CLK output for MII PHY modes of operation.

2.2.3.1 MII MAC Mode

The MII MAC Mode, sometimes called 'Forward MII', configures Port 9 or Port 10's MAC inside the devices to act as a MAC so it can be directly connected to an external MII-based PHY. In this mode, the devices receive the interface clocks (Px_TXCLK and Px_RXCLK) from the PHY and will work at any frequency from DC to 50 MHz (50 MHz supports 200 Mbps in each direction). The two clocks can be asynchronous with each other. Both full- and half-duplex modes are supported and need to be selected to match the mode of the link partner's MAC. The MII MAC mode is compliant with IEEE 802.3 clause 22. (**Note:** The MII requires only four data bits in each direction so only the lower four data bits are used). P9_MODE or P10_MODE should be set correctly at reset (see Table 8 and Table 10) to select this configuration and the PHY's SMI address must be set to 0x09 for Port 9 or 0x0A for Port 10 for auto-negotiation to operate correctly.

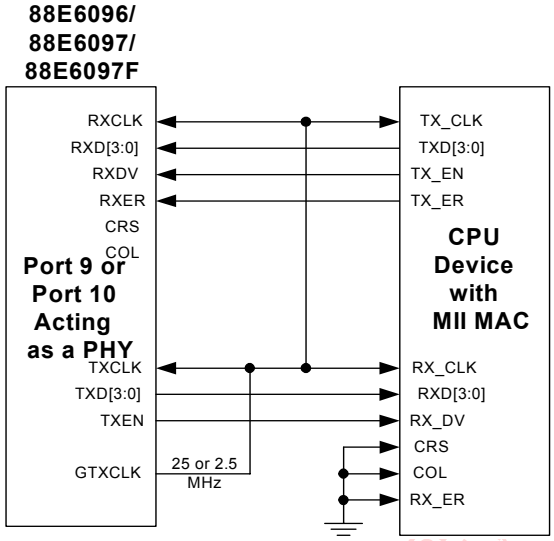
Figure 8: MII MAC Interface Pins



2.2.3.2 MII PHY Mode

The MII PHY Mode, sometimes called 'Reverse MII', configures Port 9 or Port 10's MAC inside the devices to act as a PHY so that it can be directly connected to an external MAC. In this mode, the devices drive the interface clocks (RXCLK and TXCLK for both MACs) from its GTXCLK pin so the appropriate GTXCLK frequency must be selected. For the 88E6096/88E6097 devices, the GTX_CLK pin is only available on Port 10. The GTX_CLK pin is available on both Port 9 and Port 10 of the 88E6097F device. GTX_CLK is used as a generic asynchronous clock source, but it is recommended that there are not more than four loads on GTX_CLK. For connection to more than four loads (that is, connection for use with Port 9), buffer Port 10s GTX_CLK, or use a generic oscillator. Only full-duplex modes are supported (since CRS and COL are not driven by the devices outputs) and must match the mode of the link partner's MAC. The MII PHY mode is compliant with IEEE 802.3 clause 22 in full-duplex mode (**Note**: The MII requires only four data bits in each direction so only the lower four data bits on the devices are used). At reset, P9_MODE and P10_MODE should be set for the appropriate speed —see Table 8 and Table 10. In this mode, there is no external PHY for Port 9 or Port 10, and so Port 9 or Port 10 is skipped by the PPU. In Reverse MII mode initially, the link status is down requiring the system software to force the port's link up to enable the port.

Figure 9: MII PHY Interface Pins

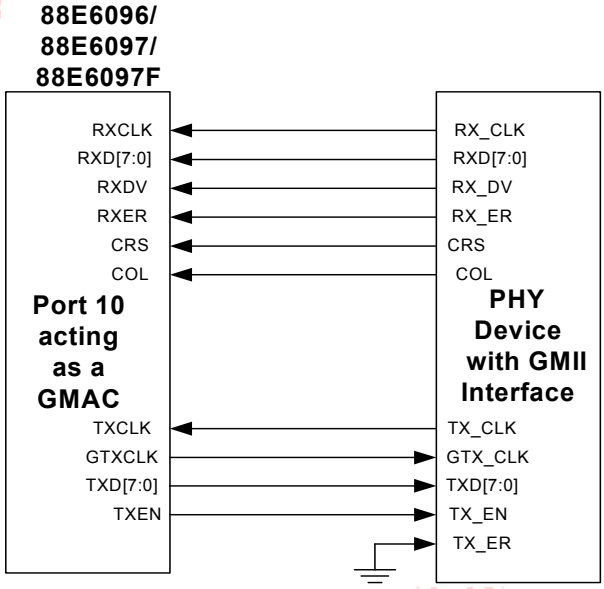


2.2.3.3 GMII MAC Mode

The GMII MAC Mode, sometimes called 'Forward GMII', configures Port 10's MAC (also Port 9's MAC for the 88E6097F device) inside the devices to act as a gigabit MAC (GMAC) so that it can be directly connected to an external GMII-based Gigabit PHY. In this mode, the devices receive the interface clocks (TXCLK and RXCLK) from the PHY but generates GTXCLK for the PHY. 10 Mbps, 100 Mbps or 1000 Mbps is supported in this configuration. Full- and half-duplex modes are supported at 10 Mbps or 100 Mbps. Full-duplex is supported at 1000 Mbps. The speed and mode in the external PHY's auto-negotiation must be restricted from advertising the 1000BASE, half-duplex case as the GMAC inside the devices do not support that mode. This is done automatically by the PHY Polling Unit (PPU) inside the devices. GMII MAC mode is compliant with IEEE 802.3 clause 28. P9_MODE and P10_MODE should be set to GMII mode at reset (see Table 8 and Table 10) for this configuration and the PHY's SMI address must be set to 0x09 for Port 9 or 0x0A for Port 10 for auto-negotiation to operate correctly.

A triple speed interface is supported in GMII MAC mode (i.e., 10, 100 and 1000). When the PHY completes auto-negotiation and brings the link up the auto-negotiated speed, duplex and flow control information must be moved from the PHY to the MAC so the MAC matches the PHY's settings. This is done automatically by the PPU if the port's PHYDetect bit is set to a one (Port offset 0x00). The interface pins will track the speed that the MAC is set to.

Figure 10: GMII MAC Interface Pins

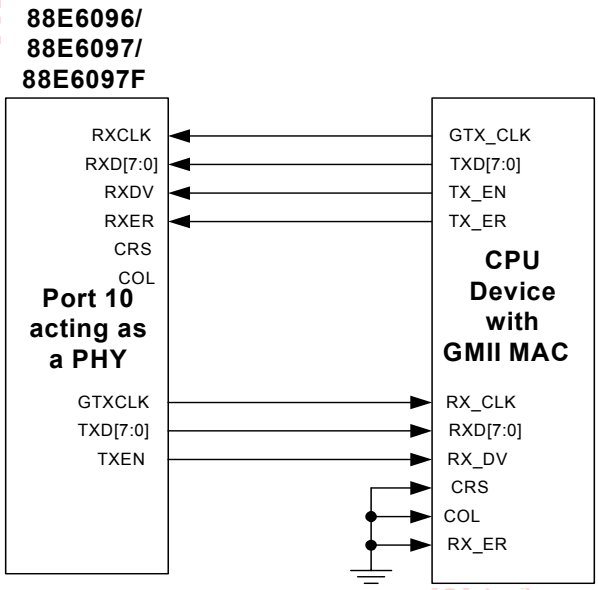


2.2.3.4 GMII PHY Mode

The GMII PHY Mode, sometimes called 'Reverse GMII', configures Port 10's MAC (also Port 9's MAC for the 88E6097F device) inside the devices to act as a gigabit PHY so that it can be directly connected to an external GMAC. In this mode, the devices drive the transmit interface clock (GTXCLK) and accept the receive interface clock (RXCLK). Only gigabit full-duplex mode is supported and must match the mode of the link partner's GMAC. GMII PHY mode is compliant with IEEE 802.3 clause 28 in gigabit full-duplex. P9_MODE and P10_MODE should be set to GMII mode at reset (see Table 8 and Table 10). In this mode, there is no external PHY for Port 9 or Port 10, so Port 9 or Port 10 are skipped by the PHY Polling Unit (PPU). Initially, the link status is configured down requiring the system software to force the port's link up to enable the port (in the PCS Control Register).

This configuration is identical to the GMII MAC Mode described above except that a CPU is connected instead of a PHY. The lack of an external PHY device restricts the interface to a gigabit speed only with the link initially being down. This allows the CPU time to initialize itself before it enables the switch port connected to it by forcing link up in the switch port's MAC (in the port's PCS Control Register - offset 0x01).

Figure 11: GMII PHY Interface Pins

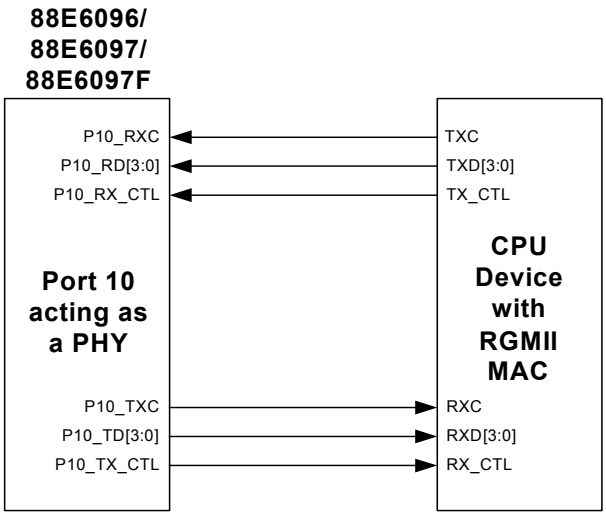


2.2.3.5 RGMII Mode

The RGMII Mode configures Port 10's MAC to act as a Reduced Gigabit Media Independent Interface (RGMII) so that it can be directly connected to an external RGMII-based Gigabit MAC inside a CPU. When the RGMII mode is selected, transmit control (P10_TX_CTL) is presented on both clock edges of P10_TXC. Receive control (P10_RX_CTL) is presented on both clock edges of P10_RXC.

The lack of an external PHY device restricts the interface to a gigabit speed only with the link initially being down. This allows the CPU time to initialize itself before it enables the switch port connected to it by forcing link up in the switch port's MAC (in the port's PCS Control register - offset 0x01).

Figure 12: RGMII Interface Pins



2.2.4 PHY Polling Unit (PPU)

The devices contain a PHY Polling Unit (PPU) to transfer Link, Speed, Duplex and Pause information from an external PHY to its associated MAC (the internal PHYs use a direct approach such that this information is transferred even if the Port's PHYDetect bit is zero - Port offset 0x00). The PPU can perform this job only if the SMI address of the external PHY matches the physical port number it is connected to in the switch (i.e., the PHY connected to Port 8 uses SMI address 0x08, the PHY connected to Port 9 uses SMI address 0x09, and the PHY connected to Port 10 uses SMI address 0x0A, etc.).

If the PPU is disabled on a port (i.e., the port's PHYDetect bit is zero), software must perform the job of setting the switch MAC's mode to the mode of the PHY (for the external PHYs) by forcing the MAC's link, speed, duplex and pause settings (in the port's PCS Control Register - offset 0x01) based upon what it sees in the PHY's registers. Link up must be the last mode register set and link down must be the first mode register cleared (i.e., the port's speed, duplex and pause modes must only be changed while the port's link is down).

Note

Even though the PPU has full access to the external and internal PHY's registers, software can access all of the PHY registers at any time by using the SMI Command and Data registers (Global 2, offsets 0x18 and 0x19). On previous pin compatible devices, the PPU could be disabled by a PPUEn bit (Global 1, offset 0x04) and the PPU's state could be read in the PPUState bits (Global 1, offset 0x00). These bits exist and function the same in this device, but they are documented differently or as Reserved bits as this is no longer the recommended way to access the PHY registers (use the SMI Command and Data registers instead - Global 2, offsets 0x18 and 0x19). The PPUEn and PPUState bits will physically be changed in future devices to match this documentation.

2.2.5 24 Port Switch Configuration

The P9_TXD[3]/SW_24P configuration pin can be used to pre-configure Port 8 and Port 9 for a three-chip 24 port switch configuration without the need of a CPU or EEPROMs. This mode is intended for unmanaged designs only and should not be used if the design is different from Figure 4.

If SW_24P is pulled high at the rising edge of RESETn Ports 8 and 9 are effected in the following way:

- They are configured as Distributed Switching Architecture (DSA) Tag ports
- They are forced to an cross-chip interface (C_MODE = 0x4 - Table 72)
- Link is forced up, Duplex is forced to full-duplex and speed is forced to gigabit
- They are restricted from sending frames to each other (i.e., their VLANTable is set to 0x4FF on both ports)

The MDC_PHY/LRN2ALL configuration pin performs LRN2ALL configuration.

Section 3. Switch Core Functional Description

The device has been designed for many different applications. For flexibility, and to facilitate learning how to use the switch, all switch ports have been designed with identical capabilities as far as the switch core is concerned[1]. At the same time, the physical port speed options and connections to the outside world are different depending upon the port number (see Section 2).

This section focuses on the central switch core functions that are identical for all the ports. While the identical port capabilities make the device flexible, it may be confusing which features should be used in a given mode and/or application. To help understand how best to use the features of the device based upon application, the Switch Core Functional Description is separated into the following parts:

- Basic Switch Functions – those functions that are common to all Frame Modes (Section 4).
- Normal Network Frame Mode – for IEEE standard untagged and tagged frames or for 'customer' ports on switches with at least one 'provider' port (Section 5).
- Provider Frame Mode – for IEEE standard 'provider' ports (Section 6).
- Distributed Switch Architecture (DSA) Frame Mode – for multiple chip switch fabric extensions or for connections to a switch management CPU and/or Router CPU (Section 7).

Each switch port can be in one of the three basic modes of operation:

- Normal
- Provider
- DSA

Where two DSA options are supported:

- Classic
- Ether type

The port modes of operation are configured using the port's FrameMode register (Port offset 0x04).

Notes

- For purpose of discussion, the word "device" refers to the 88E6096 device, the 88E6097 device, and the 88E6097F device, unless stated otherwise.
- Device specific features that refer to the 88E6097 device also refer to the 88E60967F device.

1. There is one exception: Remote Management (Section 10) is not supported on all ports for security reasons.

Section 4. Basic Switch Functions

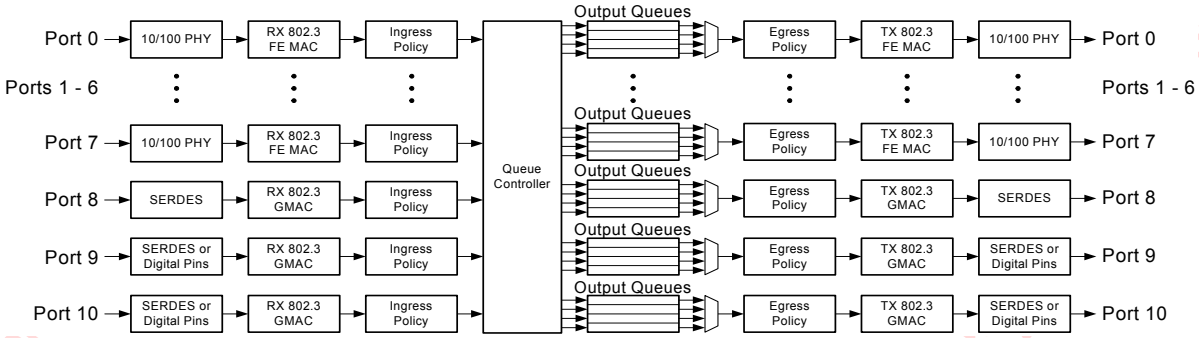
The following basic switch operations occur on all ports regardless of the port's FrameMode (Port offset 0x04).

4.1 Physical Switch Data Flow

The device accepts Ethernet frames and either discards them or transmits them out of one or more of the switch's ports. The decision on what to do with each frame is just one of the many tasks handled inside the switch. Figure 13 shows the data path inside the switch along with the major functional blocks that process the frame as it travels through the device. Each of these blocks along with their register-controllable options and policy is described in the following sections.

This section focuses on the frame processing and policies that take place in the switch core (from MAC receive to MAC transmit) of a single port.

Figure 13: Switch Data Flow



4.2 Physical Interface

Each port contains a physical interface of some sort to receive and transmit frames to and from the port's MAC. Some ports support many different physical interface options while others support only one. If a port supports many interface options only one option can be used at a time. The physical interface options that each port supports are covered in Section 2.

Note

Device features are discussed with references to the registers that control the features. The registers in the switch device are organized into three groups called Port, Global 1 and Global 2 with an additional group used to access the PHYs called PHY. Each of these groups support up to 32 16-bit registers and each port has its own set of 32 Port registers. A specific register out of the 32 in a group can be referred to by the term 'offset'. For example, the Port Control register is referenced as Port offset 0x04 as it appears in the Port device address space at register address 0x04. See Section 13 for details on the registers.

4.3 Media Access Controllers (MAC)

The device contains eight independent 10/100 MACs and three gigabit MACs (triple speed GMACs). These MACs perform all of the functions in the 802.3 protocol including frame formatting, frame stripping, CRC checking, CSMA/CD enforcement, and collision handling. All MACs (including the GMACs) support 10/100 Mbps operation in either full-duplex or half-duplex mode. Each GMAC supports 1 Gbps operation in full-duplex mode only (Ports 8, 9 and 10). For the rest of this section 'MAC' generically refers to the 10/100 MACs on Ports 0 to 7 or to the GMACs on Ports 8 to 10.

The MAC receive block checks incoming packets and discards those with CRC errors, those with alignment errors, short packets (less than 64 bytes), or long packets (more than 1522 bytes)[1]. Each MAC constantly monitors it's receive lines waiting for preamble bytes followed by the Start of Frame Delimiter (SFD). The first six bytes after the SFD are used as the packet's Destination Address (DA)[2] and the next six bytes after that are used as the packet's Source Address (SA). These two addresses are fundamental to the operation of the switch (see Section 4.4 for more information). The next two to sixty bytes are examined and may be used for QoS (Quality of Service) or policy decisions made by the switch (see Section 5 for more information). The last four bytes of the packet contain the packet's Frame Check Sequence (FCS). The FCS must meet the IEEE 802.3 CRC-32 requirements for the packet or it will be discarded.

Before a packet can be sent out, the transmit block must check if the line is available for transmission. The transmit line is available all the time when the port is in full-duplex mode, but the line could be busy receiving a packet if the port is in half-duplex mode. If the line is busy, the transmitter waits by deferring its transmission. When the line is available the transmitter ensures that a minimum interpacket gap of at least 96 bits occurs prior to transmitting a 56-bit preamble and an 8-bit Start of Frame Delimiter (SFD) ahead the frame. Actual transmission of the frame begins immediately after the SFD.

For half-duplex mode, the device also monitors the collision signal while it is transmitting. If a collision is detected (i.e., both transmitter and receiver of a PHY are active at the same time) the MAC transmits a JAM pattern and then delays the re-transmission for a random time period determined by the IEEE 802.3 backoff algorithm. In full-duplex mode, the collision signal and backoff algorithm are ignored.

4.3.1 Backoff

In half-duplex mode, the device's MACs implement the truncated binary exponential backoff algorithm defined in the IEEE 802.3 standard. This algorithm starts with a randomly-selected small backoff time and follows by generating progressively longer random backoff times. The random times prevent two or more MACs from always attempting re-transmission at the same time. The progressively longer backoff times give a wider random range at the expense of a longer delay, giving congested links a better chance of finding a winning transmitter. Each MAC in the device resets the progressively longer backoff time after 16 consecutive retransmit trials when the DiscardExcessive bit is cleared to a zero (Global 1 offset 0x04). Each MAC then restarts the backoff algorithm with the shortest random backoff time and continues to retry and retransmit the frame. A packet that successively collides is re-transmitted until transmission is successful. This algorithm prevents packet loss in highly-congested environments. The MACs in the switch are configured to meet the IEEE 802.3 specification and discard a frame after 16 consecutive collisions instead of restarting the backoff algorithm when the DiscardExcessive bit is set to a one (Global 1 offset 0x04).

1. A maximum frame size of 1632 bytes is supported by setting the MaxFrameSize bit in the Global Control register (Global 1, offset 0x04).
2. The first six bytes of a frame are processed as the frame's DA unless the Marvell® Header mode is enabled on the port (Port offset 0x04). If the Marvell Header mode is enabled the first two bytes are processed as the Marvell Header and the next six bytes are processed as the frame's DA.

4.3.2 Half-duplex Flow Control

Half-duplex flow control is used to throttle the throughput rate of an end station to avoid dropping packets during network congestion. It is enabled on all half-duplex ports via the EE_DIN/HD_FLOW_DIS pin (see Table 16). Flow control can be enabled or disabled on any particular port by forcing the port's Flow Control mode (FCValue and ForcedFC in the PCS Control Register, Port offset 0x01). The device uses a mixed carrier assertion and collision based scheme to perform half-duplex flow control. When the free buffer space is almost empty, the MAC issues carrier and/or collision which prevents further incoming packets. Only the ports that are involved in the congestion are flow controlled. If the half-duplex flow control mode is not set and there is no packet buffer space available, the incoming packet is discarded.

Half-duplex flow control is not an IEEE defined feature. The IEEE defined full-duplex flow control is described in the next section.

4.3.3 Full-duplex Flow Control

IEEE 802.3 flow control mechanism requires two link partners to auto-negotiate and advertise their flow control capabilities. If both link partners are flow control capable, then flow control will be enabled in both link partners MACs. The PHYs are used to advertise the capability but the flow control itself is a function of the MAC. The IEEE flow control also requires full-duplex operation.

The purpose of full-duplex flow control is the same as in half-duplex – avoid dropping packets during congestion. If the full-duplex flow control mode is not set and if there is no packet buffer space available, the incoming packet is discarded.

Full-duplex flow control is enabled on all full-duplex ports via the EE_CLK/FD_FLOW_DIS pin (see Table 16), if Auto-Negotiation is enabled on the port, and if the link partner 'advertises' that it supports Pause during Auto-Negotiation. Basically, full-duplex flow control is automatically enabled on a port if:

- The port's PHY is advertising it supports flow control.

and

- The port's PHY sees that its link partner is also advertising it supports flow control too (once link is established).

The EE_CLK/FD_FLOW_DIS pin (at the rising edge of RESETn) determines the initial flow control advertisement bit setting in the PHYs of this device. The inverted value of this pin is moved to external PHYs by the PPU (Section 2). Internal PHYs will have their flow control advertisement bit initialized (to the inverted value of this pin) even if the PPU is disabled.

When flow control is enabled using the EE_CLK/FD_FLOW_DIS or EE_DIN/HD_FLOW_DIS device pins, it is enabled for all ports of the same type (i.e., on all half-duplex ports or on all full-duplex ports that have a flow-controllable link partner). It may be required to have flow control enabled on only one or two ports and disable flow control on all other ports. In this case, flow control should be disabled via FD_FLOW_DIS and HD_FLOW_DIS device pins, which will disable flow control on all the ports. The ports chosen to have flow control enabled can then be configured to advertise flow control. This can be done by writing to the internal or external PHYs flow control advertisement bit (by using the SMI PHY Command and Data registers – Global 2 offsets 0x18 and 0x19). The PPU must be enabled on the port (the port's PHYDetect bit equal to one - Port offset 0x00) to allow the MAC to determine the flow control results of auto-negotiation with the link partner.

In full-duplex mode, the device's MACs support flow control as defined in the IEEE 802.3 standard. This flow control mechanism enables stopping and restarting packet transmission at the remote node. The basic mechanism for performing full-duplex flow control is via a Pause frame. The format of the Pause frame is shown in Table 23.

.

Table 23: Format of Pause

Destination Address (6 Bytes)	Source Address (6 Bytes)	Type (2 Bytes)	Op Code (2 Bytes)	Pause Time (2 Bytes)	Padding (42 Bytes)	FCS (4 Bytes)
01-80-C2-00-00-01	See text	88-08	00-01	See text	All zeros	Computed

Full-duplex flow control works as follows. When a port's free buffer space is almost empty the device sends out a Pause frame with the maximum pause time to stop the remote node from sending more frames into the switch. Only the ports that are involved in the congestion are Paused. When congestion on the port is relieved, the device sends out a Pause frame with pause time equal to zero, indicating that the remote node may resume transmission.

The device also responds to the Pause command in the MAC receiving block. When the Pause frame is detected, the port responds within one slot time to stop transmission of new data for the amount of time defined in the pause time field of the received Pause frame.

The Source Address of a received Pause frame is not learned[1] since it may not represent the Source Address of the transmitting port. This is generally the case if the link partner is an unmanaged switch. The Source Address of transmitted Pause frames can be configured (see switch MAC address register, Global 2 offset 0x0D). A single fixed Source Address can be used for all ports, or a unique Source Address per port can be selected by the changing the value of the DiffAddr bit in the switch MAC Address register.

The MACs discard all IEEE 802.3 Pause frames received. This is always the case, even if full-duplex flow control is disabled or if the port is in half-duplex mode.

4.3.4 Forcing Flow Control in the MAC

Section 4.3.3 describes the IEEE defined flow control mechanism, which requires auto-negotiation with a link partner. Some ports may not have a PHY attached, or there may be a PHY attached without auto-negotiation. In this case, flow control can be enabled or disabled by forcing the port's Flow Control mode (FCValue and ForcedFC in bit in the port's PCS Control Register, Port offset 0x01). Forcing flow control in this way will instruct the port's MAC to transmit Pause frames when needed and act on received Pause frames. It does not change the advertisement bits in the port's PHY[2].

If the port has a PHY connected (either internal or external) with auto-negotiation enabled, it is best to not force flow control (by using FCValue and ForcedFC) if the port is in full-duplex mode. Instead set the PHY's auto-negotiation flow control advertisement bit to allow flow control to occur automatically if the port's link partner agrees.

4.3.5 Jamming Control - Egress Limit

Perfect flow control, i.e. no packet loss, (full- or half-duplex) can cause network problems. A potential problem can occur between two switch boxes that simultaneously Pause each other off at exactly the same time such that neither can drain their full buffers. This very rare, but possible, situation can cause a dead-lock on the link between the two switch boxes. It is easily solved by limiting the number of back-to-back Pause refreshes a port can transmit and thus the maximum time the link partner can be stalled, allowing the dead-lock to clear. The Port's LimitOut register (Port offset 0x02) controls the number of maximum Pause times that a port can stall its link partner. The

1. See Automatic Address Learning in Section 4.4.3.
2. In this case the port's link partner may not be supporting Pause frames because it does not see from the PHY that this port is advertising it wants to support Pause.

range of the LimitOut register is large enough to ensure zero packet loss under normal, or even extreme, network congestion situations while at the same time ensuring a dead-lock situation does not occur.

Note

1 maximum Pause time is 65,536 slot times where 1 slot time is 512 bit times. A bit time is 100 ns for a 10 Mbps port, 10 ns for a 100 Mbps port and 1ns for a Gigabit port. Therefore, 1 maximum Pause time is 33.55 mSec at Gigabit, 335.5 mSec at 100 Mbps and 3.355 secs at 10 Mbps.

4.3.6 Jamming Control - Ingress Limit

When flow control is enabled on a port, it can be stalled by its link partner such that the port can never transmit any frames. This could be a result of the problem described above (Section 4.3.5) or it could be a DoS attack (Denial of Service). The port's LimitIn register (Port offset 0x03) can be used to limit how long a port's egress queue can be jammed. Once the programmed limit is reached, flow control will be forced off on the port and an interrupt generated to the CPU (if enabled – Global 2 offset 0x00 and 0x01). Software can determine which port reached the limit by examining the ports flow control forcing bits. Flow control will be forced off on any port whose limit was reached (ForcedFC will = 1 and FCValue will = 0 in Port offset 0x01). Software can re-enable flow control on the port changing the value of these bits (by clearing the port's ForcedFC bit to zero).

If the port is in full-duplex mode the automatic disabling of flow control on the port will allow frames to egress the port once the last Pause time has expired. But if the port is in half-duplex mode, constant collisions from the link partner can still prevent frames to egress the jammed port. For this reason, the Jam Limit interrupt will be activated on half-duplex ports even if flow control is disabled on these ports. Software can take action to either enable DiscardExcessive (see Section 4.3.1 or Global 1 offset 0x04) or to Disable or Block the port (see Port States in Port offset 0x04). In either case, the goal is to free up the jammed buffers for other ports to use (see Section 5.6).

4.3.7 Forcing Link, Speed, and Duplex in the MAC

Link, Speed and Duplex can be forced on a port's MAC by using the port's ForcedLink, ForcedDpx, and ForceSpd registers (Port offset 0x01). Extreme caution must be used when forcing one of these modes on a port's MAC. For example: Do not change the MAC's Speed or Duplex unless the port's Link is down.

These bits change the port's MAC mode only! It does not change the mode of the PHY for ports where a PHY is connected. These bits are intended to be used for the following situations only:

- When the PHY Polling Unit (PPU) is disabled on the port (PHYDetect equal to zero in Port offset 0x00) and software needs to copy the PHY's Link, Speed and Duplex values to the port's MAC (this is not required for internal PHYs as this information is communicated between the PHY and MAC even if the PPU is disabled on the port).
- When no PHY is connected to the port. This includes ports that connect to a CPU (typically using a digital interface like MII or GMII) and ports connected to another switch device (typically using a SERDES interface). SERDES ports connected to a fiber module will get their Link from the port's SDET pin and its Speed and Duplex is set to 1000BASE full-duplex (assuming the Px_MODE has been set correctly – see the C_Mode bits, Port offset 0x00).

Ports without PHYs attached will have their Link down until software forces the port's Link up. The Speed and Duplex of these ports should not be forced as the hardware strapping on the Px_MODE pins will set the Speed and Duplex on these links.

4.3.8 MAC Based RMON/Statistics Counters

The MAC Based Statistics Counter logic maintains a set of 28, 32-bit counters and two 64-bit counters per port, that enable the user to monitor network performance remotely and to support RMON groups 1, 2, 3, and 9. These counters provide a set of Ethernet statistics for frames received on ingress and transmitted on egress. Switch Policy Statistics counters are also supported. See Section 4.3.9.

The counters are designed to support:

- RFC 2819 – RMON MIB (this RFC obsoletes 1757 which obsoletes 1271)
- RFC 2665 – Ethernet-like MIB (this RFC obsoletes 1643, 1623 and 1398)
- RFC 2233 – MIB II (this RFC obsoletes 1573 & 1213 with obsoletes 1229 & 1158)
- RFC 1493 – Bridge MIB (this RFC obsoletes 1286)

The complete description of each of the counters is contained in Table 24 and Table 25.

All CPU register interfaces are slow compared with the speed of Gigabit or even Fast Ethernet frames. For this reason all the RMON counter data associated with a port can be placed into an Ethernet frame and transmitted to the CPU (or other device). Two options are supported, a MIB Dump and a MIB Dump and Clear. See Remote Management described in Section 10.

Alternately, the device supports a snapshot function to capture instantly and hold static any port's MAC Statistics counters (see the Stats Operations register, Global 1 offset 0x1D, for more information). The capture function maintains a higher level of accuracy between the various counters in a port and also allows multiple counter values to be added together to get the required MIB. After capture, the CPU can take its time to read out the values of the counter or counters that it needs without concern for the values changing during the register read process.

The CPU interface supports the following operations on the Stat Counters:

- Clear all counters for all ports
- Clear all counters for a single port
- Capture all counters for a single port
- Read a captured counter (a Capture must be executed before a Read to the capture zone can be done)
- Read a counter directly (best used when reading only one counter on a port)

See the Stats Operation Register (Global 1 offset 0x1D) for more details.

Note

The Set 4 counters can be configured to be ingress only, egress only, or both.

Table 24: Ingress Statistics Counters

Name	Offset Address	Description
Set 1		
InGoodOctetsLo	0x00	The lower 32-bits of the 64-bit InGoodOctets counter. The sum of lengths of all good Ethernet frames received, that is frames that are not bad frames.
InGoodOctetsHi	0x01	The upper 32-bits of the 64-bit InGoodOctets counter. See description above.
InBadOctets	0x02	The sum of lengths of all bad Ethernet frames received.
Set 2		
InUnicast	0x04	The number of good frames received that have a Unicast destination MAC address.
InBroadcasts	0x06	The number of good frames received that have a Broadcast destination MAC address.
InMulticasts	0x07	The number of good frames received that have a Multicast destination MAC address. **Note:** This does not include frames counted in InPause nor does it include frames counted in InBroadcasts.
InPause	0x16	The number of good frames received that have a Pause destination MAC address.
Set 3		
InUndersize	0x18	Total frames received with a length of less than 64 octets but with a valid FCS.
InFragments	0x19	Total frames received with a length of less than 64 octets and an invalid FCS.
InOversize	0x1A	Total frames received with a length of more than MaxSize octets but with a valid FCS.
InJabber	0x1B	Total frames received with a length of more than MaxSize octets but with an invalid FCS.
InRxErr	0x1C	Total frames received with an RxErr signal from the PHY.
InFCSErr	0x1D	Total frames received with a CRC error not counted in InFragments, InJabber or InRxErr.
Set 4		**These counters can be Ingress Only, Egress Only, or both**
64Octets	0x08	Total frames received (and/or transmitted) with a length of exactly 64 octets, including those with errors.
65to127Octets	0x09	Total frames received (and/or transmitted) with a length of between 65 and 127 octets inclusive, including those with errors.

Table 24: Ingress Statistics Counters (Continued)

Name	Offset Address	Description
128to255Octets	0x0A	Total frames received (and/or transmitted) with a length of between 128 and 255 octets inclusive, including those with errors.
256to511Octets	0x0B	Total frames received (and/or transmitted) with a length of between 256 and 511 octets inclusive, including those with errors.
512to1023Octets	0x0C	Total frames received (and/or transmitted) with a length of between 512 and 1023 octets inclusive, including those with errors.
1024toMaxOctets	0x0D	Total frames received (and/or transmitted) with a length of between 1024 and MaxSize[1] octets inclusive, including those with errors.

1. MaxSize is 1522 for non-tagged frames and for tagged frames if MaxFrameSize = 0 or MaxSize = 1632 if MaxFrameSize = 1 (Global 1 offset 0x04).

Table 25: Egress Statistics Counters

Name	Offset Address	Description
Set 5		
OutOctetsLo	0x0E	The lower 32-bits of the 64-bit OutOctets counter. The sum of lengths of all Ethernet frames sent from this MAC.
OutOctetsHi	0x0F	The upper 32-bits of the 64-bit OutOctets counter. See description above.
Set 6		
OutUnicast	0x10	The number of frames sent that have a Unicast destination MAC address.
OutBroadcasts	0x13	The number of good frames sent that have a Broadcast destination MAC address.
OutMulticasts	0x12	The number of good frames sent that have a Multicast destination MAC address. **Note:** This does not include frames counted in OutPause nor does it include frames counted in OutBroadcasts.
OutPause	0x15	The number of Flow Control frames sent.
Set 7		
Deferred	0x05	The total number of successfully transmitted frames that experienced no collisions but are delayed because the medium was busy during the first attempt. This counter is applicable in half-duplex only.
Collisions	0x1E	The number of collision events seen by the MAC not including those counted in Single, Multiple, Excessive, or Late. This counter is applicable in half-duplex only.
Single	0x14	The total number of successfully transmitted frames that experienced exactly one collision. This counter is applicable in half-duplex only.
Multiple	0x17	The total number of successfully transmitted frames that experienced more than one collision. This counter is applicable in half-duplex only.
Excessive	0x11	The number frames dropped in the transmit MAC because the frame experienced 16 consecutive collisions. This counter is applicable in half-duplex only and only of DiscardExcessive is a one (in Switch Global Control - global offset 0x04).
Late	0x1F	The number of times a collision is detected later than 512 bits-times into the transmission of a frame. This counter is applicable in half-duplex only.
OutFCSErr	0x03	The number of frames transmitted with an invalid FCS. Whenever a frame is modified during transmission (e.g., to add or remove a tag) the frame's original FCS is inspected before a new FCS is added to a modified frame. If the original FCS is invalid, the new FCS is made invalid too and this counter is incremented.

Table 25: Egress Statistics Counters (Continued)

Name	Offset Address	Description
Set 4		**These counters can be Ingress Only, Egress Only, or both**
64Octets	0x08	Total frames transmitted (and/or received) with a length of exactly 64 octets, including those with errors.
65to127Octets	0x09	Total frames transmitted (and/or received) with a length of between 65 and 127 octets inclusive, including those with errors.
128to255Octets	0x0A	Total frames transmitted (and/or received) with a length of between 128 and 255 octets inclusive, including those with errors.
256to511Octets	0x0B	Total frames transmitted (and/or received) with a length of between 256 and 511 octets inclusive, including those with errors.
512to1023Octets	0x0C	Total frames transmitted (and/or received) with a length of between 512 and 1023 octets inclusive, including those with errors.
1024toMaxOctets	0x0D	Total frames transmitted (and/or received) with a length of between 1024 and MaxSize[1] octets inclusive, including those with errors.

1. MaxSize is 1522 for non-tagged frames and for tagged frames if MaxFrameSize = 0 or MaxSize = 1632 if MaxFrameSize = 1 (Global 1 offset 0x04).

4.3.9 Policy Based RMON/Statistics Counters

The device maintains a set of policy counters (one 32-bit and two 16-bit) per port that enable the user to monitor network performance by seeing where good frames have been dropped by the switch (bad frames that are dropped are counted in the MAC based counters – Section 4.3.8). Some frames are dropped due to switch policy and others are due to excessive congestion in the switch.

The policy counters are:

- InDiscards - A 32-bit counter (16 bits in InDiscardsLo, Port offset 0x10, and 16 bits in InDiscardsHi, Port offset 0x11) that counts the number of good, non-filtered frames that normally would have been forwarded, but could not be due to a lack of buffer space.
- InFiltered - A 16-bit counter (Port offset 0x12) that counts the number of good frames that were filtered due to ingress switch policy rules. These rules include frames that are dropped due to Layer 2 Policy Control Lists (PCLs, Port offset 0x0E), 802.1X MAC authentication (SA Filtering - Port offset 0x04), MAC Address Learn Limiting (Port offset 0x0C), 802.1Q Security checks (802.1QMode Port offset 0x08), DiscardTagged & DiscardUntagged (Port offset 0x08), PortState other than Disabled (Port offset 0x04), and DA mappings back to the source port (normal switch filtering).
- OutFiltered - A 16-bit counter (Port offset 0x13) that counts the number of good frames that were filtered due to egress switch policy rules. These rules include frames that passed the ingress port's policy but are dropped due to the egress policy of this port including 802.1Q Security checks (802.1QMode Port offset 0x08) if NoEgrPolicy is zero (Global 2, offset 0x1D) and PortState other than Disabled (Port offset 0x4).

These counters stop counting when the port's PortState is set to Disabled (Port offset 0x04) and they are all cleared when a Flush All Counters for this port or a Flush All Counter for All Ports command is issued to the MAC based counters (see the Stats Operation Register, Global 1 offset 0x1D, for more details).

All CPU register interfaces are slow compared with the speed of Gigabit or even Fast Ethernet frames. For this reason, all the RMON counter data associated with a port can be placed into an Ethernet frame and transmitted to the CPU (or other device). Two options are supported, a MIB Dump and a MIB Dump and Clear. See Remote Management described in Section 10.

4.4 Basic Switch Operation

The switch portion of the device receives good packets from the MACs, processes them, and forwards them to the appropriate MACs for transmission. The primary task of the switch is to process frames and this activity involves the following blocks shown in Figure 13.

- Ingress Policy (Section 5.1)
- Queue Controller (Section 5.6)
- Output Queues (Section 5.6.5.2)
- Egress Policy (Section 5.8)

These blocks modify the normal or default packet flow through the switch.

The normal packet flow and processing through a switch involves learning how to switch packets to the correct MACs, and only to the correct ones. The switch learns what port an end station is connected to by remembering each packet's Source Address[1] along with the port number on which the packet arrived. Once a MAC address/port number mapping is learned, all future packets directed to that end station's MAC address (as defined in a frame's Destination Address field) are directed to the learned port number only. If a packet is directed to a new, currently unlearned, MAC address, the packet will be transmitted out of all the ports[2], except for the one on which it arrived[3]. This ensures that the packet is received by the correct end station (if it exists) and when the end station responds back its address is learned by the switch for the next series of packets.

All switches learn only a very small subset of the set of possible MAC addresses owing to the limits of physical memory. Switches learn only the currently 'active' MAC addresses. Sometimes end stations are moved from one port to another so that a new MAC address/port number association must be learned and the old one replaced. All of these issues are handled by what is called 'Aging' and 'Station Move Handling'. Basically, a MAC address/port number association is allowed to be 'active' for only a limited amount of time. This time limit is typically set to five minutes.

The following sections describe how the device performs its basic switch functions.

4.4.1 Lookup Engine

The device's Lookup Engine or Address Translation Unit (ATU) uses the DA and SA fields from each frame received from each port. It performs all address searching, address learning, and address aging functions for all ports at 'wire speed' rates (i.e., a DA and an SA lookup/learn function can be performed for all ports in less time than it takes to receive a 64 byte frame on any port).

The address database uses a hashing technique for quick storage and retrieval. Hashing a 48-bit address into fewer bits results in some MAC addresses having the same hash address. This is called a hash collision and is solved in the device by using four bins per hash location allowing for storage of up to four MAC addresses at each hash location. This allows the address database to be smaller while still holding the same number of active, random value MAC addresses.

The address database is stored in embedded SRAM and has a size of 8192 entries with a default aging time of about 300 seconds or 5 minutes. The age time can be modified in 15 second increments from 0 seconds (aging disabled) to 3825 seconds (almost 64 minutes). These options are set in the ATU Control register (Global 1 offset 0x0A).

1. The SA on switch management frames (Section 8.2) are not learned. This includes the IEEE Pause frame.
2. VLANS modify this operation – see Section 5.2 and Section 5.2.2.
3. The device can be configured to transmit frames out the port they came in on – see Section 5.3.1.

4.4.2 Address Searching or Translation

The address search engine is used to search the address database to get the output port number(s), called the Destination Port Vector (DPV), for each frame's destination address so that it can switch the frame instead of flooding[1] it. It arbitrates destination address lookup requests from the ports and grants one lookup at a time. The address is hashed and then data is read from the SRAM table, looking for a MAC address match. Four addresses can be stored at each hash location. If a match is found, the Address Translation Unit (ATU) returns the Destination Port Vector (DPV) to the Ingress Policy block where it may be modified[2] before the packet is queued to the output ports. If the found entry contains a Trunk ID, the Trunk ID is converted to a DPV using the Trunk Mapping Table (Global 2, offset 0x08). If no MAC address match is found the Ingress Policy block uses a unique default DPV for each ingress port[3], which typically floods the frame. If the destination address in the frame is a multicast address or broadcast address, the address is searched[4] in the same way as a unicast address and the frame is processed identically. This feature is used for multicast filtering. Multiple separate address databases are supported in the device. The database that is searched is controlled by the port's default Forwarding Information Database (FID in the Port Based VLAN Map register, Port offset 0x06, and the Port Control 1 register, Port offset 0x05) or the one assigned to the frame by the VTU (Section 9.2.1) based on the VID assigned to the frame during ingress (Section 5.2.2). MAC addresses that are not members of the port's or frame's FID cannot be found.

4.4.3 Automatic Address Learning

The address learning engine is used to learn the source address of ingressing frames. Up to 8192 MAC address/port number mappings can be stored in the address database (see Section 4.4.1 for more information). When the source address from an input frame can not be found in the address database, the ATU enters the self-learning mode and places the new MAC address/port number mapping into the database and refreshes its Age time[5]. If the MAC address is found to be already in the database, the port information and Age associated with the entry is updated and/or refreshed. The port number/Trunk ID is updated in case the end station moved and the port number or Trunk ID needs to be corrected. The entry's Age is refreshed since the MAC address is still 'active'. This prevents the MAC address/port number mapping from being removed as being 'inactive' prematurely.

When an address is added into the database it is hashed and stored in the first empty bin found at the hashed location. If all four address bins are full, a least recently used algorithm is used for looking at each entry's Age time (its EntryState field[6]). If all four address bins have the same Age time, then the first 'non-static' bin is used (see Section 9.1.1 for more information about locked or static addresses). If all four bins are 'static' the address is not learned and an ATUFull interrupt is generated (see the Switch Global Status register, Global 1 offset 0x00). The port information stored with the new MAC address is the port's Port Association Vector (PAV, Port offset 0x0B) if the source port is not a Trunk port (Trunk Port bit Port offset 0x05) or it is the port's Trunk ID (Port offset 0x05) if the source port is a Trunk port.

Multiple separate address databases are supported in the device (see Section 4.4.8). The port's Forwarding Information Database (FID in the Port Based VLAN Map register, Port offset 0x06, and the Port Control 1 register, Port offset 0x05) determines into which address database the MAC address is added if 802.1Q is disabled on the port. If 802.1Q is enabled on the port the FID associated with the frame's VID (VLAN ID field of Tagged frames, see Section 9.2.1) determines the address database into which the MAC address is stored. The same MAC address can be learned multiple times with different port mappings if different FID values are used.

1. Flooding refers to the action of sending frames out all the ports of the switch except for the port the fame came in on.
2. The DPV returned from the ATU may be modified by the VLANTable data, VTU results, the Trunk Mask Table, and/or other filters.
3. The default DPV for each port is the list of ports that can egress multicast frames, if the frame is multicast or the list of ports that can egress unicast frames, if the frame is unicast (Egress Floods in Port Control register, Port offset 0x04). Broadcast frames are considered multicast frames unless Flood BC is set to a one (Global 2, offset 0x05).
4. Multicast addresses cannot be auto learned. Multicast addresses must be loaded manually with a CPU or EEPROM.
5. The Age time on a MAC Address entry is refreshed by setting its EntryState field to 0x7- see Table 42.
6. The EntryState field is described in Section 9.1.1.

Learning can be disabled on any individual port by clearing the port's PAV to all zeros (see Port Association Vector, Port offset 0x0B) or by setting the port's Learn Disable bit to a one (in Port Based VLAN Map register, Port offset 0x06). Learning is also disabled on any port that has a PortState of Disabled or Blocking/Listening (see the Port Control register, Port offset 0x04).

Learning is never performed on switch management frames. This includes Pause frames (Section 4.3.3), BPDU, LAC and other management frames as long as they are determined to be MGMT frames (see Management frames - Section 8.2), and Distributed Switching Architecture (DSA) Tag frames (Section 7) with the exception of the Forward type of DSA Tag frames.

4.4.4 Hardware Address Learn Limiting

Automatic address learning can be limited by hardware independently per port in the range from 1 to 255 addresses. This feature is enabled by setting the LearnLimit register (Port offset 0x0C) to the desired limit. Once enabled, the port's learn counter (LearnCtr Port offset 0x0C) keeps track of the number of MAC addresses learned by incrementing once each time a new MAC address is learned on the port (i.e., an address that was not already present in the address database). When the learn counter reaches the Learn Limit value the Limit Reached bit is set to a one (Port offset 0x0C) and one of two events will occur with subsequent frames that enter this port:

1. Frames containing a Source Address (SA) in the address database that is associated with the port the frame entered will be allowed to enter the port[1]. This is true for all MAC addressed in the address database, including those already in the database prior to the Learn Limit being enabled, and those added by the CPU either as static or aging.
2. Frames containing an SA that is not in the address database or one that is in the address database, but is not associated with the port the frame entered, will be discarded. If the frame contains a new SA an ATU Miss Interrupt will be generated (see Section 9.1.6) if the port's Over Limit Int En bit is set to a one (Port offset 0x0C).

The learn counter will decrement by one[2] whenever an address associated with this port ages out of the address database (see Section 4.4.5). In this case the port's Limit Reached bit will be cleared indicating the counter is below the limit. Now if a frame with a new Source Address enters the port, the frame will be accepted and its SA will be learned because the limit counter is less than the Learn Limit.

Only automatic operations affect the port's learn counter (LearnCtr). CPU ATU operations such Load and Purge (see Section 9.1.3) do not effect the port's learn counter. The only exceptions to this are the ATU Flush All entries and the ATU Flush All Non-Static entries commands. Since both of these operations clear out all non-static entries, all port's learn counters are reinitialized to zero.

A port's LearnCtr will be reinitialized to zero whenever the port's Learn Limit is set to zero (i.e., whenever the port's learn limit function is disabled).

To get accurate results, it is best to enable the learn limit function before frames are allowed to flow into the port (i.e., when the port is in the Disabled or Blocking Port State, Port offset 0x04). If the port's learn limit needs to be changed to a larger number after frames are allowed to flow, the LearnLimit can be increased at any time. But if the port's learn limit needs to be changed to a smaller number after frames are allowed to flow the following procedure must be followed:

1. Disable learning on the port. Either clear the port's PAV (Port offset 0x0B) or set the port's LearnDisable bit (Port offset 0x06).
2. Clear out all addresses associated with this port in the address database. Either issue an ATU Flush All Non-Static or do an ATU Move Non-Static to port 0xF (Global 1 offset 0x0B).

1. The entry's Age will be refreshed as well if it is not static (see Section 4.4.5)
2. The decrement is clipped at zero to cover the case where addresses were already present in the address database, associated with this port, prior to the Learn Limit being enabled.

3. Clear the port's LearnLimit to zero to reinitialize the port's LearnCtr (this is not needed if the ATU Flush All Non-Static operation was used above).
4. Set the port's LearnLimit to the new value.
5. Re-enable learning on the port by reversing what was done in step 1 above.

Notes

- Hardware Address Learn Limiting requires that Learn2All (Global 1 offset 0x0A) must be set to a one and that Locked Port is cleared to zero on the port (Port offset 0x0B) and that the port is not a member of a Trunk (Trunk Port is cleared to zero in Port offset 0x05).
- If either the source port and/or the destination port of a station move[1] has hardware Address Learn Limiting enabled, the station move will not take place. This a is self correcting situation as the station move will take place as soon as the MAC address of the station move ages out (Section 4.4.5) as it will if the station actually moved.

4.4.5 Automatic Address Aging

The address aging process is used to ensure that if a node is disconnected from the network segment, or if it becomes inactive, its entry is removed in a timely manner from the address database. Aging makes room for new active addresses. An address is removed from the database after a programmable amount of time from the last time it appeared in an ingressing frame's Source Address. This programmable time is determined by the Age Time bits in the ATU Control register (Global 1 offset 0x0A).

The device runs the address aging process continuously (unless disabled by setting the AgeTime field to zeros). Aging is accomplished by a periodic sweeping of the address database. The speed of these sweeps determines the aging time. On each aging sweep of the database, the ATU reads each valid entry and updates its age time by decrementing its EntryState field[2] (as long as the entry is not static – see Section 9.1.1). When the EntryState field reaches zero, the entry is considered invalid and purged from the database. The EntryState field will not decrement past 0x1 (i.e., it will not be automatically purged) if the port's HoldAt1 bit is set (Port offset 0x0B). HoldAt1 is intended to be used with CPU directed Address Learning (Section 4.4.6).

A new or just refreshed unicast MAC address has an EntryState value of 0x7 (see Section 4.4.3). A purged or invalid entry has an EntryState value of 0x0. The values from 0x6 to 0x1 indicate the Age time on the unicast MAC address with 0x1 being the oldest. This scheme results in seven age states on an entry allowing the Address Learning's least recently used replacement process (Section 4.4.3) to be more precise. An address is purged from the database within 1/7th of the programmed AgeTime value making the address's lifetime interval in the database more accurate as well.

4.4.6 CPU Directed Address Learning and Purging

Sometimes it is required to prevent automatic learning from occurring on a port and have a CPU direct the learning instead. The device supports CPU directed learning on a per port basis by setting the port's LockedPort bit to a one (in the Port Association Vector register – Port offset 0x0B). When a port is 'Locked' all frames received with an SA not found in the address database cause an SA Miss ATU Violation as long as learning is enabled on the port (i.e., the port's PAV, offset 0x0B, is non-zero). The SA Miss ATU Violation can be set to generate a hardware interrupt—see the ATUProbIntEn bit of the Switch Global Control register (Global 1 offset 0x04). One ATU Violation per port is held in the ATU. Once a violation is captured all subsequent violations are ignored until the first one is serviced by the CPU.

1. A station move occurs when the Source Address (SA) on a frame is found in the address database but the database's port information on that address does not match the port where the frame came from.
2. The EntryState field is described in Section 9.1.1.

The CPU can retrieve the source MAC address and the source port information of the SA Miss Violation by issuing an ATU Get/Clear Violation Data ATUOp (Section 9.1.6). The CPU then decides if the address should be placed into the address database or not. If it should be, the CPU issues a Load ATUOp (Section 9.1.3).

If the CPU loads the new address as 'non-static', the entry stays in the address database until it ages out. Its age time is determined by the loaded EntryState value[1], as addresses on Locked ports do not have their age time refreshed, unless the port's RefreshLocked bit is set to a one (Port offset 0x0B). In this case, addresses already present in the address database will be automatically refreshed (i.e., get their EntryState set to 0x7) as long as the association on the address is not changing (i.e., as long as it is not a station move). Learn2All message frames are generated on these automatic refreshes if the Learn2All bit is set to a one (Global 1 offset 0x0A).

The CPU receives no new interrupts from non-static addresses until they age out or are purged out by the CPU, unless the ATUAgeIntEn bit is set to a one (Global 2 offset 0x5) and/or unless the port's IntOnAgeOut bit is set to a one (Port offset 0x0B). If the global ATUAgeInt is enabled the CPU will receive an ATU Miss Violation if the EntryState found in the address database on the ingressing frame's SA is less than 0x4 (and the port's RefreshLocked bit is cleared to zero). If the port's IntOnAgeOut (interrupt on age out) is enabled, the CPU will receive an Age Out Violation whenever any address associated with the port[2] is at an EntryState of 0x1 when aging starts to process the entry. The entry will then either be aged out of the address database, or its EntryState value will be held a 0x1 if the port's HoldAt1 bit is set (Port offset 0x0B). The Hold at 1 feature requires that the CPU to purge the entry from the address database so the CPU can control when and where addresses are removed.

If the CPU loads the new address as 'static', the entry stays in the address database until the CPU purges it. In this case, the CPU receives no new interrupts from this address as long as the address is never used as an SA on a port other than the original source port. If the MAC address is used on a different source port, the CPU can receive an ATU Member Violation interrupt if the IgnoreWrongData bit (see the Port Association Vector register, Port offset 0x0B) is cleared on the port where the frame just entered the switch. This interrupt may indicate that a station move just occurred or that an end station is masquerading by using another station's address.

4.4.7 802.1X Source MAC Address Checking

The device supports 802.1X source MAC address authentication using CPU Directed Address Learning (Section 4.4.6) along with the DropOnLock Ingress policy (Section 5.1.2). CPU Directed Address Learning is required for 802.1X[3] so that the requesting MAC address can be authorized by the authorization server. The DropOnLock policy causes all frames from unauthorized MAC addresses to be discarded.

A side effect of authorization is that a CPU might become saturated from constant SA Miss Violations from a source that it has denied. This is prevented in the device by masking denied MAC addresses. An address can be masked by loading it into the address database with a Destination Port Vector (DPV) of all zeros[4]. The address appears in the database so it no longer causes a 'Miss' Violation. Any port trying to send a frame to this unauthorized address is discarded (since its DPV is all zeros) and any 802.1X port trying to use this unauthorized address has its frames discarded too (since the port's SA bit is not set in the ATU entry). But now the CPU will get SA Member Violation interrupts every time the unauthorized address is attempted to be used as an SA. These interrupts can be masked too by setting the IgnoreWrongData bit (Port Association Vector – Port offset 0x0B) on the 802.1X ports.

The CPU can mask unauthorized MAC addresses by loading them into the address database as static or non-static entries. If they are loaded non-static, the interrupts are masked until the entry ages out of the database or until the CPU purges the entry (do not enable the port's RefreshLocked feature, Port offset 0x0B, in this case as

1. The Age Time (Global 1, offset 0x0A) determines the age time as well. See Section 4.4.5.
2. The association may be direct from the entry's DPV or indirect from the entry's Trunk ID mapped through the Trunk Mapping Table (Global 2 offset 0x08).
3. An 802.1X port needs to have its LockedPort bit set to one and its SAFiltering bits set to 0x1 (see the Port Association Vector, Port offset 0x0B, and the Port Control register, Port offset 0x04).
4. The all zero DPV cannot be used to mask these interrupts if the enhanced 802.1X Drop to CPU mode is being used to trap semi-authorized frames to the CPU (Section 5.1.2.2).

the refresh will 'authorize' the MAC address by updating the entry's DPV). This approach minimizes the number of interrupts the CPU needs to service while keeping the address database fluid. If the unauthorized address is loaded as static, the interrupts are masked until the CPU purges the entry. This requires the CPU to remember addresses it has loaded because at some time the CPU should purge these addresses to make room for new ones. The use of too many static addresses can also cause an ATU Full Violation, so it is best to mask addresses using the non-static approach. The DropOnLock feature prevents the reception of frames from all unauthorized MAC addresses regardless of whether the unauthorized address is currently being masked in the address database or not. The masking feature is intended to minimize the number of SA Miss Violation interrupts the CPU needs to service for addresses that it already has denied.

If a port is saturating a CPU by constantly using a new SA, the masking of unauthorized addresses is not applicable. Instead all SA Miss Violations can be masked on a port by disabling learning on the port (i.e., by setting the port's Port Association Vector bits to all zero (Port offset 0x0B). This configures the ingress port in a form of a Secure Port (Section 5.1.3) and will work as long as no new SA needs to be authorized on this port.

4.4.8 Multiple Address Database Support (FID)

The device supports up to 4,096 separate and independent address databases in the Address Translation Unit (ATU). Multiple address database are used to isolate MAC addresses by VLAN so the same MAC address can appear multiple times in the address database with different port mappings. The device uses a Forwarding Information Database (FID) mechanism as defined in 802.1Q to isolate the address databases from each other. Although the address database is isolated by FID value it is not divided up in equal segments by FID value. Each database number can hold none to all of the possible 8192 MAC addresses or any number in between. Any number of FID values can be used (from 1 to 4,096). Each forwarding information database (FID) uses only the MAC address entries it needs and leaves all the remaining ATU entries for all the other databases.

Each frame, as it ingresses a port, is assigned a FID. The frame's FID value, along with the frame's DA and SA, is sent to the Address Translation Unit (ATU) when the frame's MAC addresses are searched and/or learned. The frame's FID is determined in priority order by:

- The FID associated with the frame's VID[1] in the VLAN Translation Unit (VTU, Section 9.2.1). This requires the frame's VID to be valid in the VTU. All frames are assigned a VID, even untagged frames. This is true if 802.1Q is enabled on the port or not.
- The FID associated with the ingressing port (FID[11:4] in Port Control 1 register, Port offset 0x05 and FID[3:0] in Port Based VLAN Map register, Port offset 0x06).

If multiple address databases are not needed leave all the FID values at their reset value of 0x000 in all their occurrences in the registers (in the ports, ATU and VTU).

A frame's FID is generally determined by the frames VID (via the VTU, Section 9.2.1). Multiple VID's can be mapped to the same FID allowing for shared VLAN address databases.

1. The frame's VID can be overridden. See Section 5.2.2.7.

Section 5. Normal Network Ports

The discussions that follow assume the port is in Normal Network mode, unless specified otherwise (i.e., Frame-Mode = 0x0 at Port offset 0x04).

5.1 Ingress Policy

The Ingress Policy block is used to modify the normal packet flow through the switch, generally by limiting where they are allowed to go using industry standard mechanisms. All ports have identical capabilities.

- Non-management frame types can be blocked from entering the switch to support Spanning Tree Protocol (Section 5.1.1 for classic 802.1D and Section 5.2.3 for 802.1s).
- The frame's source MAC address can be authenticated and the frame potentially discarded or mapped to the CPU for 802.1X support (Section 4.4.7 and Section 5.1.2).
- Layer 2 Policy actions (mirror, trap or discard) can be performed based on the frame's DA, SA, VID, Ether type, and/or if the frame is a UDP broadcast and/or a DHCP Option 82 (88E6097 only – Section 5.1.3).
- Port based VLANs and/or 802.1Q VLANs are used to prevent a frame from going out of certain ports, and switch management Port States, 802.1s (per VLAN spanning tree) or 802.1Q are used to prevent the frame from entering the switch at all (Section 5.2).
- All IEEE 802.1Q Tagged frames can be discarded (Section 5.2.2.4) or all Untagged frames can be discarded (Section 5.2.2.3).
- Port based Ingress Rate Limiting (PIRL) is supported with five (88E6097 only) or two independent counters/ resources per port (Section 5.5).
- Any Layer 2 Policy mirror (88E6097 only – Section 5.1.3) and any Ingress Monitor Source mirror (Section 8)[1] can be statistically sampled using PIRL by mirroring only 1 of every 'n' frames (88E6097 only – Section 5.5).

5.1.1 Port States Filtering for 802.1D Spanning Tree

The device supports four 802.1D Port States per port shown in Table 26 (802.1s per VLAN Port States are also supported - see Section 5.2.3). The 802.1D Port States are used by the Queue Controller (see Section 5.6) in the device to adjust buffer allocation. They are used by the Ingress Policy blocks to control which frame types are allowed to enter and leave the switch so that Spanning Tree or bridge loop detection software can be supported. The PortState bits in the Port Control register (Port offset 0x04) determine each port's Port State (assuming 802.1s per VLAN Port States are not being used) and the bits can be modified at any time without causing any errors on transmitted frames.

Table 26 lists the Port States and describes them. Two of the Port States require the detection of management (MGMT) frames. MGMT frames are defined in Section 8.2. Their primary purpose is to support the Spanning Tree Protocol (see Section 8.2) so these frames have the ability to tunnel through blocked ports. SA learning is not performed on MGMT frames. MGMT frames also ignore VLAN rules on ingress and egress (802.1Q and Port Based), IGMP snooping and Rate Limiting (if MGMT frames are selected for non-rate limiting in PIRL – Section 5.5). This means they always go to the port indicated by the Destination Port Vector (DPV) assigned to the frame's DA in the address database or to the device's CPUDest port (Global 1 offset 0x1A) depending upon what MGMT detection mode was used (Section 8.2). MGMT frames are typically used for 802.1D Spanning Tree Bridge Protocol Data Units (BPDUs), but any multicast or any unicast address can be used supporting new or proprietary protocols.

1. ARP Mirrors (Section 5.3.3) cannot be statically sampled.

Table 26: Port State Options

Port State	Description
Disabled	Frames are not allowed to enter (ingress) or leave (egress) a Disabled port. Learning does not take place on Disabled ports.
Blocking/Listening	Only MGMT frames are allowed to enter (ingress) or leave (egress) a Blocked port. All other frame types are discarded. Learning is disabled on Blocked ports.
Learning	Only MGMT frames are allowed to enter (ingress) or leave (egress) a Learning port. All other frame types are discarded but learning takes place on all good non-MGMT frames that are not discarded owing to being filtered[1].
Forwarding	Normal operation. All frames are allowed to enter (ingress) and leave (egress) a Forwarding port. Learning takes place on all good non-MGMT frames that are not discarded owing to being filtered[1].

1. Frames can be filtered for many reasons including Layer 2 Policy (Section 5.1.3), Port States (Section 5.1.1 and Section 5.2.3), 802.1X MAC Authentication (Section 5.1.2.1), 802.1Q Violations (Section 5.2.2), reverse 802.1X MAC Authentication (Section 5.1.2.3) or its Tagging did not match what was expected (Section 5.2.2.3 and Section 5.2.2.4).

The default Port State for all the ports in the switch can be either Disabled or Forwarding depending upon the value of the SW_MODE pins (Table 17). The ports come up in the Forwarding Port State unless the SW_MODE is configured for CPU attached mode. This allows the CPU to fully boot before it brings up the ports and starts accepting frames including running bridge loop or spanning tree detection or routing software.

Notes

- The internal and external PHYs will reset in the powered down state whenever the SW_MODE pins are configured for CPU attached mode. Software must power up the PHYs before they will link by clearing the PHY's PwrDwn bit to zero (PHY offset 0x00). Software only needs to power up the PHYs once after booting. This ensures that the PHY's link partners do not see link until the CPU has had time to boot and get ready to accept MGMT frames from its link partner.
- Managed switches must always use the CPU attached mode setting of the SW_MODE pins. But even these designs may require a jumper or test point on the PCB such that the switch can be made to reset to the EEPROM attached mode setting of the SW_MODE pins. This can help initial PCB debug and/or manufacturing test of the switch portion of the design as the switch will power up forwarding frames everywhere without any software, but this is a debug mode only and must not be used in production designs where a CPU is attached.

5.1.2 Source Address Filtering

The device supports Source Address (SA) filtering for 802.1X MAC Authentication, Trapping frames to the CPU for further inspection prior to fully authenticating the SA if desired, or Reverse 802.1X MAC Authentication to help prevent Denial of Service (DoS) attacks. This is accomplished by loading specific addressed into the address databases along with per port mode bits.

5.1.2.1 802.1X MAC Address Authentication (Drop on Lock)

All non-MGMT[1] (non-management) frames received on a port with an unauthorized Source MAC Address are discarded if the port's SA Filtering bits are set to 0x1, Drop on Lock mode (Port offset 0x04). In this mode, only frames with an authorized SA (or MGMT frames) are allowed into the switch to be processed further. An SA is

1. See Section 8.2 on how to detect a MGMT frame.

considered authorized if it is present in the address database and its contents are associated to the source port where the frame entered the switch[1]. The policy of how these addresses are entered into that address database is controlled by the ATU (Section 9.1). It is recommended that CPU Directed Address Learning (Section 4.4.6) be used on ports supporting MAC based 802.1X authentication (i.e., ports in DropOnLock mode). Before the CPU authenticates an address it may want to inspect the contents of some frames. This can be done using Source MAC Frame Trapping (Section 5.1.2.2). Excessive interrupts from denied Source Addresses can be masked in this mode (see Section 4.4.7).

5.1.2.2 Source MAC Frame Trapping (Drop to CPU)

Source MAC Frame Trapping works with 802.1X MAC Address Authentication (Section 5.1.2.1) and it is enabled on a port if the port's SA Filtering bits are set to 0x3, Drop to CPU mode (Port offset 0x04). The purpose of this mode is to get the CPU more data before it authenticates the Source MAC. In standard 802.1X the CPU has to authenticate the SA based on the SA only. This mode allows the trapping of frames from the requesting SA to the CPU for further inspection. Now the CPU can look at the contents of entire frames to help make the authentication decision.

Drop to CPU mode works identically to the Drop on Lock mode (Section 5.1.2.1) with one difference. It adds the concept of a semi-authorized MAC address where frames with a semi-authorized SA are mapped to the CPU's port (based on the value of CPUDest, Global 1, offset 0x1A). This means that all non-MGMT[2] (non-management) frames received on a port with an unauthorized Source MAC Address are discarded. In this mode, only frames with an authorized or semi-authorized SA (or MGMT frames) are allowed into the switch to be processed further. An SA is considered authorized if it is present in the address database and its contents are associated to the source port where the frame entered the switch[3]. An SA is considered semi-authorized if it is present in the address database and its Destination Port Vector (DPV and 'T' bit) are all zeros. This all zeros value ensures that no other port can transmit frames to this SA (as frames using this MAC address as a DA will be discarded due to the all zero DPV) while allowing frames with that SA to be sent to the CPU for further inspection.

The policy of how these addresses are entered into that address database is controlled by the ATU (Section 9.1). It is recommended that CPU Directed Address Learning (Section 4.4.6) be used on ports supporting MAC based 802.1X authentication (i.e., ports in DropOnLock mode).

- Drop to CPU frames egress the CPU's port as a non-MGMT frame and they are filtered based upon the egress port's Port State (Port offset 0x04). If the CPUDest port on the device is using DSA tags (Section 6) the Drop to CPU frames will egress as DSA Forward frames. In multi-chip or stacking systems, all switch devices in the path from the original ingress port to the actual CPU must have the semi-authenticated MAC addresses entered into their address database or the CPU will not get these trapped frames. These MAC addresses can be entered with SA Priority Override if needed (Section 5.4.6).
- Excessive interrupts from denied SA's cannot be supported in this mode (see Section 4.4.7). If this is needed use 802.1X Drop on Lock (Section 5.1.2.1) and then frames can be trapped to the CPU using Layer 2 Policy (88E6097 only - see Section 5.1.3).

5.1.2.3 Reverse 802.1X MAC Address Authentication (Drop on UnLock)

Reverse 802.1X MAC address authentication accepts frames from all Source MAC Addresses except for those MAC Addresses that are specifically identified to be denied. When this feature is used together with Address Learn Limiting (Section 4.4.4) or CPU Directed Learning (Section 4.4.6), they can be used to prevent Denial of Service (DoS) attacks.

1. If the port is a non-Trunk port then the ATU entry must not be a Trunk entry and it must have the port's bit set in its DPV. If the port is a Trunk port then the ATU entry must be a Trunk entry matching the Trunk ID of the ingressing port. See Section 8.10.
2. See Section 8.2 on how to detect a MGMT frame.
3. If the port is a non-Trunk port then the ATU entry must not be a Trunk entry and it must have the port's bit set in its DPV. If the port is a Trunk port then the ATU entry must be a Trunk entry matching the Trunk ID of the ingressing port. See Section 8.10.

All non-MGMT[1] (non-management) frames received on a port with a semi-authorized Source MAC Address are discarded if the port's SA Filtering bits are set to 0x2, Drop on UnLock mode (Port offset 0x04). In this mode, frames with an unknown or known SA (or MGMT frames) are allowed into the switch to be processed further. An SA is considered known if it is present in the address database and its contents are non-zero[2]. The known addresses can enter the address database through auto learning (Section 4.4.3) or by CPU directed learning (Section 4.4.6). An SA is considered semi-authorized if it is present in the address database and its Destination Port Vector (DPV and 'T' bit) are all zeros. This all zeros value ensures that no other port can transmit frames to this SA (as frames using this MAC address as a DA will be discarded due to the all zero DPV) while at the same time ensure all frames with that SA are discarded as well.

This mechanism can be used to shut down the source of a Denial of Service attack by discarding all frames coming from (and to) the identified Source Address of the attacker. If the attacker uses more than one Source Address, these too can be shut down in the same way. The number of 'denied' MAC addresses can be easily limited by combining this feature with Address Learn Limiting (Section 4.4.4) or CPU Directed Learning (Section 4.4.6) that performs the same function as Address Learn Limiting.

5.1.2.4 Static or Dynamic Addresses

When the CPU loads MAC addresses in the address database to support SA Filtering, should they be loaded as Static (non-aging) or Dynamic (aging) entries? Either can be used and it is application dependent.

- The 802.1X Drop on Lock mode cannot accept frames unless the frame's SA is in the address database associated with the ingressing port. At first look these should be loaded as static entries. But since there could be a large number of authorized MAC addresses being used at one time, it is probably better to load the authorized entries as dynamic. These addresses can be prevented from aging out if the port's HoldAt1 bit is set (Port offset 0x0B) and they can be self refreshed after being authenticated by enabling the port's Refresh Locked bit (Port offset 0x08B). If an address gets bumped out by a more recently used one, the CPU can quickly reload the already authenticated address that is needed if the CPU keeps a local cache of approved addresses. Alternatively, the CPU can perform the refreshes by configuring the device such that it gets informed about addresses that are being used whose age (or EntryState) is less than 0x4 by setting the port's IntOnAgeOut bit (Port offset 0x0B). This approach prevents the CPU from running into an ATU Full condition where a MAC address cannot be loaded (see Section 9.1.6).
- The 802.1X Drop to CPU mode should load authorized addresses as dynamic (see above), but the semi-authorized addresses must be loaded as static or only the 1st frame will be trapped to the CPU and all others will be accepted as being authorized without CPU intervention. It is assumed that the CPU will only need to look at the contents of a few frames and then it will either fully authorize the MAC or fully de-authorize it.
- The Reverse 802.1X Drop on UnLock mode allows automatic address learning with very few addresses that may need to be loaded to prevent a DoS attack. Since the number of these semi-authorized addresses is very low, the CPU can load these addresses as static. But the CPU will need to keep track of these addresses and unload them after some period of time so the port can start accepting frames again. Alternatively, the CPU can load these addresses as dynamic disallowing frames for a period of the AgeTIme (typically 5 min. – Global 1 offset 0x0A) and re-loading them as needed.

1. See Section 8.2 on how to detect a MGMT frame.
2. The contents of an ATU entry are considered zero if the entry's 'T' bit and the entry's DPV bits are zero (Section 9.1.1).

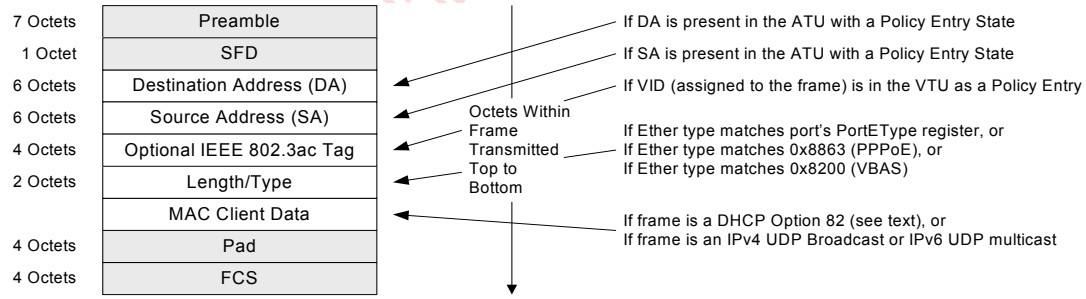
5.1.3 Layer 2 Policy Control Lists (88E6097 Only)

The device supports Layer 2 PCLs (Policy Control Lists). The policy actions supported on a per port basis and are:

- Normal frame switching (i.e., do nothing special)
- Policy Mirror (copy) the frame to the MirrorDest port[1] (Global 1 offset 0x1A)
- Policy Trap (re-direct) the frame to the CPUDest port (Global 1 offset 0x1A)
- Policy Discard (filter) the frame

The fields in the frames where the above policy is selectable on a per port basis are shown in Figure 14.

Figure 14: Fields of the Frame Examined for Layer 2 PCS



Each port supports separate actions for each of the eight possible policy items (Port offset 0x0E). Therefore, a frame could get more than one policy action applied to it. A single frame can be both Policy Trapped (to CPUDest) and Policy Mirrored (to MirrorDest), but if any policy action is Policy Discard, the frame is discarded without being mapped anywhere else. Likewise, if a frame is discarded for other switch policy reasons (like VLAN membership, Section 5.2.2, or because it is tagged or untagged, Section 5.2.2.4 and Section 5.2.2.3) the frame will not be Policy Mirrored or Policy Trapped either.

Policy Trapped frames will egress the port defined in the CPUDest register (Global 1 offset 0x1A). If this port is configured in DSA mode the frame will egress as a To_CPU frame with a CPU Code of 0x3 (see Section 7). Policy Mirrored frames will egress the port defined in the MirrorDest register (Global 1 offset 0x1A). If this port is configured in DSA mode the frame will egress as a To_CPU frame with a CPU Code of 0x5 (see Section 7).

5.1.3.1 DA, SA, and VID Policy

As each frame enters the switch, both its DA and SA are looked up into the address database. If the DA is found in the ATU with a Policy Entry State (Section 9.1.1) then Layer 2 Policy will occur on that frame on the ports where DA Layer 2 PCLs are enabled (Port offset 0x0E). If the SA is found in the ATU with a Policy Entry State then Layer 2 Policy will occur on that frame on the ports where SA Layer 2 PCLs are enabled.

As each frame enters the switch, a VID is extracted or assigned to the frame (Section 5.2.2.7) and then that VID is looked up in the VLAN database. If the VID is found in the VTU with its Policy bit set to a one (Section 9.2.1) then Layer 2 Policy will occur on that frame on the port's where VTU Layer 2 PCLs are enabled (Port offset 0x0E).

1. Any mirror in the 88E6097 can be sampled. See Section 5.5.

Notes

- Each MAC and VID entry is globally flagged as a Policy entry or not (see Section 9.1.1 for MAC and Section 9.2.1 for VID). So the same MAC address can become a Policy Discard for Ports 5 to 0, a Policy Trap for Port 6, and do normal switching for Port 7 all at the same time. But once the ports are configured this way, all MAC addresses with a Policy Entry State in the ATU will work the same way.
- DA and SA Policy entries in the ATU are static entries so auto leaning cannot change their values. The CPU will need to manually purge these entries once Layer 2 Policy is no longer needed on these addresses.

5.1.3.2 Ether Type Policy

As each frame enters the switch, its Ether type is extracted and compared. If the frame's Ether type equals 0x8863 then Layer 2 Policy will occur on that frame on the ports where PPPoE Layer 2 PCLs are enabled (Port offset 0x0e). If the frame's Ether type equals 0x8200 then Layer 2 Policy will occur on that frame on the ports where VBAS Layer 2 PCLs are enabled. If the frame's Ether type matches the port's PortEType register (Port offset 0x0F) then Layer 2 Policy will occur on that frame on the ports where EType Layer 2 PCLs are enabled.

Note

There is one programmable Ether type per port on the device that can be used for Layer 2 PCLs. This is the port's PortEType register. This register can be used for Layer 2 PCLs only if the port's Frame Mode (Port offset 0x04) is Normal Network.

5.1.3.3 DHCP Option 82

DHCP Option 82 is a special policy function that goes beyond the layer 2 fields of the frame. If the frame entering a port matches either of the frame formats shown in Figure 15 or Figure 16, then Layer 2 Policy will occur on that frame if the port's Opt82 Layer 2 PCL is enabled (Port offset 0x0E). DHCP Option 82 is supported for both IPv4 and IPv6. The portions of the frame that must match are those fields in the figures that are in a white background. Those fields in a grey background are not examined for this feature.

5.1.3.4 UDP Broadcast

UDP Broadcast is another special policy function that goes beyond the layer 2 fields of the frame. If the frame entering a port matches either of the frame formats shown in Figure 17 or Figure 18, then Layer 2 Policy will occur on that frame if the port's UDP Layer 2 PCL is enabled (Port offset 0x0E). UDP Broadcast is supported for both IPv4 and IPv6 (although in IPv6 the frame only has to be a multicast). The portions of the frame that must match are those fields in the figures that are in a white background. Those fields in a grey background are not examined for this feature.

Figure 15: IPv4 DHCP Option 82 Frame Format

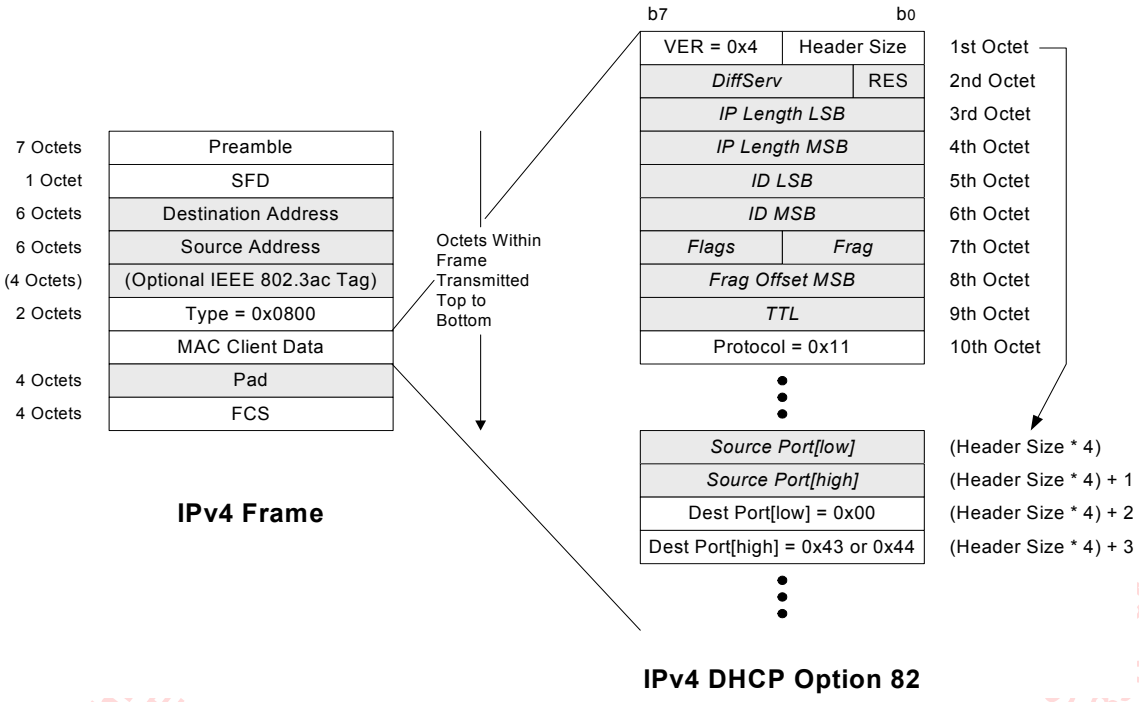


Figure 16: IPv6 DHCP Option 82 Frame Format

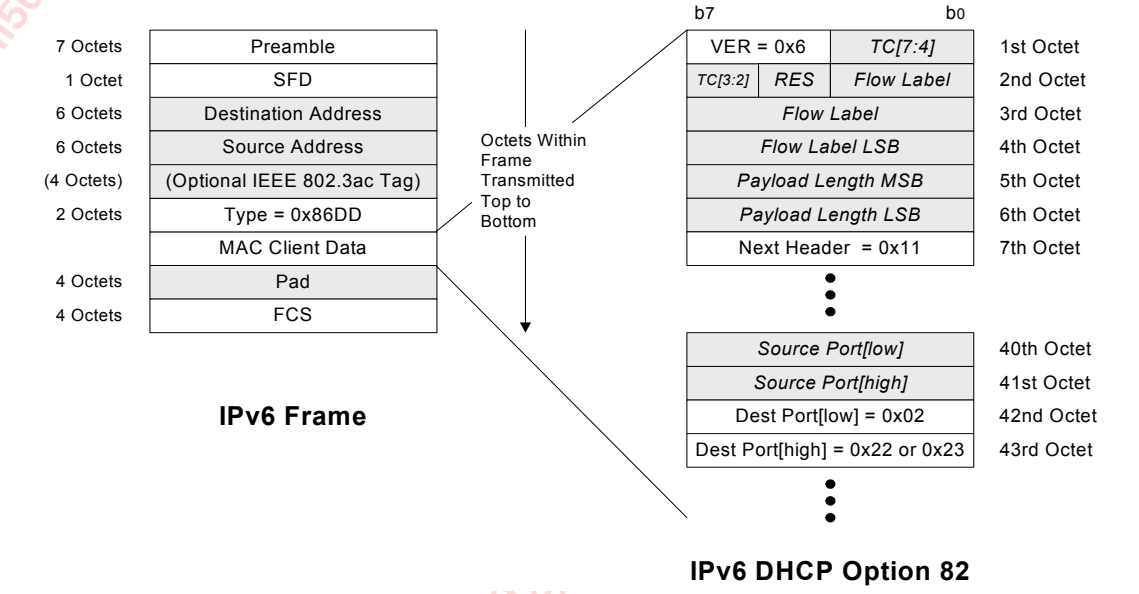


Figure 17: IPv4 UDP Broadcast Frame Format

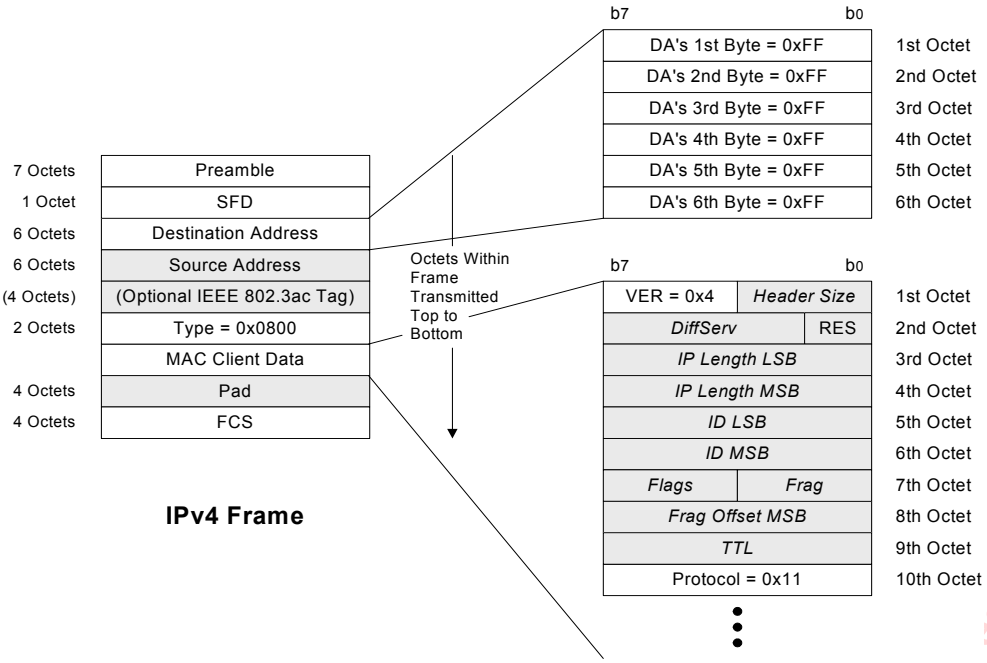
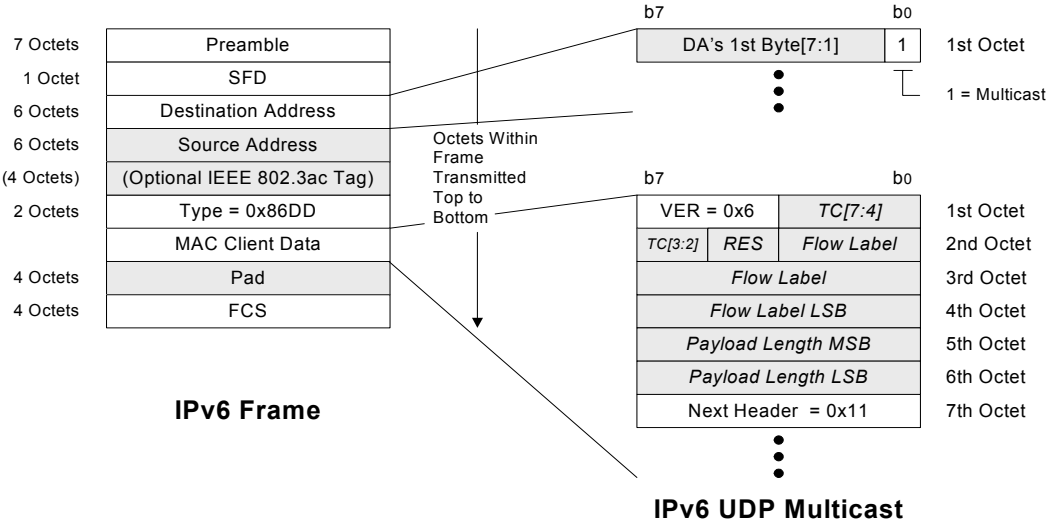


Figure 18: IPv6 UDP Multicast Frame Format



5.2 VLANS

The device supports port based VLANs and 802.1Q tag based VLANs.

5.2.1 Port Based VLANs

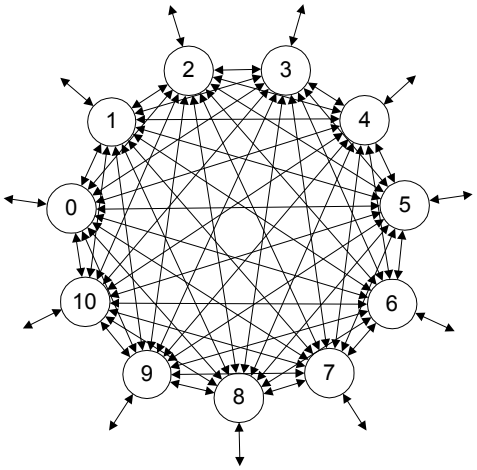
The device supports port based VLANs both in-chip (within a single device) and cross-chip (cascading across multiple devices – Section 7.5.3).

5.2.1.1 In-chip Port Based VLANs

The device supports a very flexible port based VLAN system that is used for all non-MGMT frames even if 802.1Q is enabled on the port.

Each Ingress port contains a register that restricts the output (or egress) ports to which it is allowed to send frames. This register is called the VLANTable register (Port offset 0x06). If bit 0 of a port's VLANTable register is set to a one, that port is allowed to send frames to Port 0. If bit 1 of a port's registers is set to a one, that port is allowed to send frames to Port 1. Bit 2 for Port 2, etc. At reset the VLANTable register for each port is set to a value of all one's, except for each port's own bit, which is cleared to a zero (this prevents frames from going back out of the port they came in on[1]). This default VLAN configuration allows all the ports to send frames to all the other ports as shown in Figure 19.

Figure 19: Switch Operation with Port VLANs Disabled

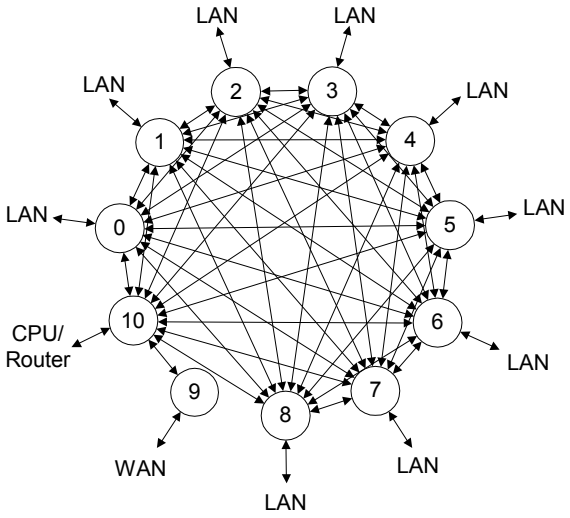


1. The device allows a port's own bit in its VLANTable to be set to a one – see Section 5.3.1

5.2.1.2 Port Based VLAN Router Examples

One of the main applications for port based VLAN support in the device is to isolate a port or ports for firewall router applications. Figure 20 shows a typical VLAN configuration for a firewall router. Port 9 is used as the WAN port. The data coming in from this WAN port must not go out to any of the LAN ports – but it must be able to go to the router CPU. All the LAN ports are able to send frames directly to each other without the need of CPU intervention – but they cannot send frames directly to the WAN port. The CPU is able to send frames to all of the ports so that routing can be accomplished. The use of the Marvell® Header[1] (Section 8.7), enables a CPU to define dynamically which port or ports a particular frame is allowed to reach for purposes of WAN and LAN isolation on multicast traffic generated by the CPU[2].

Figure 20: Switch Operation with a Typical Router VLAN Configuration



1. The Marvell Header is designed to be used on a CPU port only.
2. The Marvell Header can be used to isolate ports on multicast traffic in single chip implementations only – i.e., it is for routers.

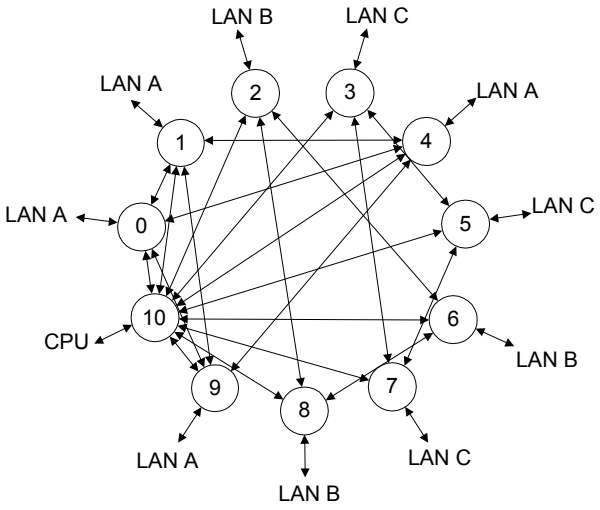
The example VLAN configuration shown in Figure 20 is achieved by setting the port's VLANTable registers as shown in Table 27:

Table 27: VLANTable Settings for Figure 20

Port #	Port Type	VLANTable Setting
0	LAN	0x5FE
1	LAN	0x5FD
2	LAN	0x5FB
3	LAN	0x5F7
4	LAN	0x5EF
5	LAN	0x5DF
6	LAN	0x5BF
7	LAN	0x57F
8	LAN	0x4FF
9	WAN	0x400
10	CPU	0x3FF

To show the flexibility of the device VLAN configuration options, Figure 21 shows another example. In this case, the switch is divided into three independent VLANs connected to a common router.

Figure 21: Switch Operation with another Example VLAN Configuration



The example VLAN configuration shown in Figure 21 is accomplished by setting the port's VLANTable registers as shown in Table 28:

Table 28: VLANTable Settings for Figure 21

Port #	Port Type	VLANTable Setting
0	LAN A	0x612
1	LAN A	0x611
2	LAN B	0x540
3	LAN C	0x4A0
4	LAN A	0x603
5	LAN C	0x488
6	LAN B	0x504
7	LAN C	0x428
8	LAN B	0x444
9	LAN A	0x413
10	CPU	0x3FF

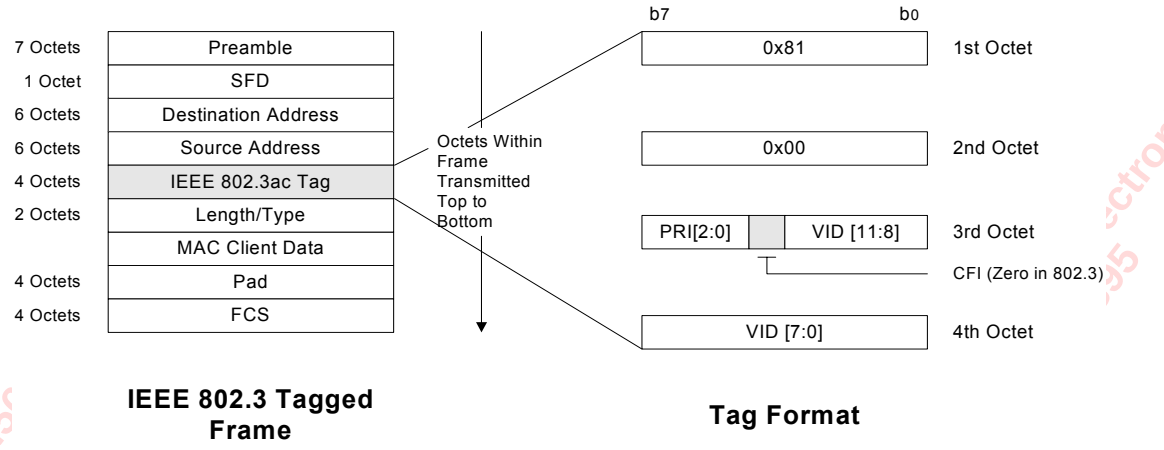
5.2.2 802.1Q VLANs

The device supports 802.1Q with the full set of 4,096 different VID (VLAN identifiers). Some or all of the VIDs can be used (i.e., software only needs to initialize the VIDs that are being used). Since the device may be programmed with only a subset of the possible VIDs, and security requirements vary, the device supports 802.1Q in three different modes. The device's port-based VLAN feature (both in-chip and cross-chip, Section 5.2.1 and Section 7.5.3) is in effect for all 802.1Q modes described below. 802.1Q VLANs are supported cross-chip as well (Section 7.5.2).

5.2.2.1 IEEE Tagging VID Handling

VLAN Identifiers (VIDs) are contained in IEEE tagged frames. All frames ingressing the device are assigned a VID, even the untagged frames. The format of an IEEE tagged frame is shown in Figure 22. The discussion below is relevant only if the port's Frame Mode is Normal Network (Port offset 0x04).

Figure 22: IEEE Tag Frame Format



5.2.2.2 Determining if a Frame is Tagged

A frame is considered physically tagged in the device if the two bytes after the frame's SA is 0x8100. A frame is considered logically tagged in the device if the two bytes after the frame's SA is 0x8100 and the tag's VID is non-zero. A physically tagged frame with a 0x000 VID is not considered logically tagged.

5.2.2.3 Discarding Untagged Frames

The device supports the discarding of frames that are not logically tagged on a port-by-port basis (see DiscardUnTagged, Port offset 0x08). A frame is considered logically untagged if the two bytes after the frame's SA does not equal 0x8100 or they do equal 0x8100 but the tag's VID equals 0x000. This is true regardless of the port's 802.1Q Mode (i.e., even if 802.1Q is Disabled on the port).

Priority only tagged frames (frames whose VID = 0x000) are considered untagged in this case and will get discarded.

5.2.2.4 Discarding Tagged Frames

The device supports the discarding of logically tagged frames on a port-by-port basis (see DiscardTagged in Port offset 0x08). A frame is considered logically tagged if the two bytes after the frame's SA equals 0x8100 and the

tag's VID does not equal 0x000. This is true regardless of the port's 802.1Q Mode (i.e., even if 802.1Q is Disabled on the port).

Priority only tagged frames (frames whose VID = 0x000) are considered untagged in this case and will not get discarded.

5.2.2.5 VID Extraction

If any of the three supported 802.1Q modes are enabled on this port (Section 5.2.2.8) the VID read from tagged frames is assigned to the frame (unless overridden – see Section 5.2.2.6). If the frame's VID = 0x000 the port's DefaultVID (Port offset 0x07) is assigned to the frame instead. If these physically tagged frames egress a port Tagged their VID bits will be overwritten with the assigned value.

Note

If 802.1Q is disabled on this port the VID bits from tagged frames are ignored, and physically tagged frames are considered physically untagged for egress tag processing (i.e., transmit Tagged will add a tag and transmit UnTagged or Unmodified will transmit the frame unmodified - Section 5.8.4). These frames are assigned the ingress port's DefaultVID (Port offset 0x07) which will be written to the frame's new (added) VID bits if the frame egresses a port Tagged.

5.2.2.6 Security Override of a Frame's VID

The device supports a VID override function where a tagged frame's VID is ignored and the port's DefaultVID is assigned to the frame instead, even if 802.1Q is enabled on the port. This is a security feature that ensures that all frames that came from a specific ingress port (tagged or untagged) exit the switch with the ingress port's DefaultVID. This prevents an end user from masquerading by simply adding an improper tag to frames.

This feature is enabled on a per port basis by setting the port's ForceDefaultVID bit to a one (Port offset 0x07).

5.2.2.7 VID Assigned to the Frame

Each frame entering the switch must have a VID (VLAN ID) assigned to it. This VID is used for 802.1Q, if enabled on the ingress port. It is also used as the frame's VID if untagged frames are to egress the switch tagged.

If a frame entering the switch is untagged, it is assigned the port's DefaultVID during Ingress (Port offset 0x07). If a frame is tagged its VID is generally used as the frame's VID unless the frame's VID is 0x000 or if the port's ForceDefaultVID is set. A summary of how a VID is assigned to each frame is shown in Table 29.

Table 29: Example VID Assignment Summary

Frame's	802.1Q Mode	Force Default VID	Default VID	Assigned VID	Comments
Don't Care	Disabled	Don't Care	0x001	0x001	Use Default VID due to 802.1Q being disabled.
0x000	Enabled	Don't Care	0x001	0x001	Use Default VID due to frame VID 0x000.
0x123	Enabled	Enabled	0x001	0x001	Use Default VID due to ForceDefaultVID = 1.
0x123	Enabled	Disabled	0x001	0x123	Use frame's VID.

5.2.2.8 Security & Port Mapping

The 802.1Q Security features of the device supports the discarding of ingressing frames that don't meet the security requirements and ensuring that those frames that do meet the requirements are sent to the allowed ports only. Three levels of security are supported and they can be set differently on each port. The security options are processed using the VID assigned to the frame (Section 5.2.2.7) as follows:

- Secure – The VID must be contained in the VTU and the Ingress port must be a member of the VLAN else the frame is discarded. The frame is allowed to exit only those ports that are both:
  - Members of the frame's VLAN
    and
  - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see Section 5.2.1)
- Check – The VID must be contained in the VTU or the frame is discarded (the frame will not be discarded if the Ingress port is not a member of the VLAN). The frame is allowed to exit only those ports that are both:
  - Members of the frame's VLAN
    and
  - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see Section 5.2.1)
- Fallback – Frames are not discarded if their VID is not contained in the VTU.
  - If the frame's VID is contained in the VTU, the frame is allowed to exit only those ports that are both:
    - Members of the frame's VLAN
      and
    - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see Section 5.2.1)
  - If the frame's VID is not contained in the VTU, the frame is allowed to exit only those ports that are:
    - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see Section 5.2.1)
- 802.1Q Disabled – Frames are not discarded if their VID is not contained in the VTU. The frame is allowed to exit only those ports that are:
  - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see Section 5.2.1)

Note

See 802.1Q Disable Mode Note in Section 5.2.2.5.

Secure, Check, Fallback, or 802.1Q Disabled modes for the port are controlled by the port's 802.1QMode bits (Port offset 0x08).

5.2.2.9 Security Violations

If 802.1Q is enabled on a port, security violations are captured and an interrupt can be generated to the CPU (if unmasked by the VTUProbIntEn bit (Switch Global Control, global offset 0x04). This is true regardless of the 802.1Q mode (VTUProb interrupts will not occur from a port if the port's 802.1QMode is 802.1Q Disabled - Port offset 0x08). The interrupts (up to one at a time) are captured by the VLAN Translation Unit (see VTU Operation register, Global 1 offset 0x05). Two kinds of security violations are captured. A MissViolation occurs if a frame's VID is not contained in the VTU. A MemberViolation occurs if a frame's VID is in the VTU but the source port of the frame is not a member of the frame's VLAN. The security violation captures the offending source port (SPID) and VID that caused the violation. This data is accessed by executing a Get/Clear Violation Data operation in the VTU.

5.2.2.10 Security Override of a Frame's VID

A Tagged frame's VID can be forced to the port's DefaultVID (see Section 5.2.2.6).

5.2.3 802.1s Per VLAN Spanning Tree

The device supports per VLAN Port States for up to 64 802.1s per VLAN Spanning Tree instances. Each VID entry in the VTU (Section 9.2.1) has a 6-bit SID value associated with it that is used to access 1 out of 64 possible 802.1s spanning tree instances from the STU (Section 9.2.6). Each STU entry contains two bits of per port 802.1s Port State information as shown in Table 30. Using the indirect STU approach allows extremely fast per VLAN spanning tree updates as only one STU entry needs to be modified for all the VIDs using the same spanning tree instance.

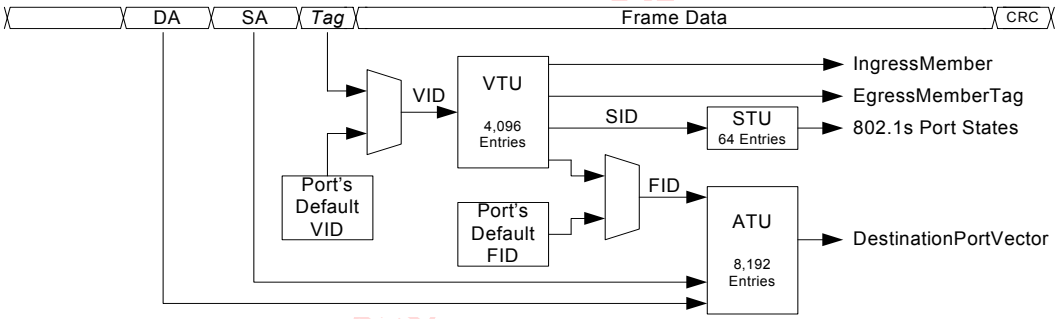
The relationship between the VTU, STU and ATU is shown in Figure 23.

Table 30: 802.1s Port State Options

Port State	Description
802.1s Disabled	The port's PortState bits in the Port Control register (Port offset 0x04) are used for this port for frames with a VID that is associated with this SID. See Section 5.1.1.
Blocking/Listening	Only MGMT frames are allowed to enter (ingress) or leave (egress) this port for frames with a VID that is associated with this SID. All other frame types are discarded. Learning is disabled on Blocked ports.
Learning	Only MGMT frames are allowed to enter (ingress) or leave (egress) this port for frames with a VID that is associated with this SID. All other frame types are discarded but learning takes place on all good non-MGMT frames that are not discarded owing to being filtered[1].
Forwarding	Normal operation. All frames are allowed to enter (ingress) and leave (egress) this port for frames with a VID that is associated with this SID. Learning takes place on all good non-MGMT frames that are not discarded owing to being filtered.

1. Frames can be filtered for many reasons including Layer 2 Policy (Section 5.1.3), Port States (Section 5.1.1 and Section 5.2.3), 802.1X MAC Authentication (Section 5.1.2.1), 802.1Q Violations (Section 5.2.2), reverse 802.1X MAC Authentication (Section 5.1.2.3) or its Tagging did not match what was expected (Section 5.2.2.3 and Section 5.2.2.4).

Figure 23: Relationship between VTU, STU, and ATU



The VTU contains a 4,096 entry database that is accessed using the VID assigned to the frame. The STU contains a 64 entry database that is accessed with SID found in the VTU. The ATU contains an 8,192 entry database that is accessed using the frame's DA with the FID found in the VTU, and using the frame's SA with the FID found in the VTU (the port's Default FID is used if 802.1Q is disabled on the port or if the VID assigned to the frame points to an empty VTU entry).

If 802.1s is not being used, all STU entries must use the 802.1s Disabled setting for all ports for all SIDs. If 802.1s is being used, all VTU entries must be configured with the required SID and each SID that is being used must define the 802.1s Port State for each port. Any SID entry can contain a mixture of ports using 802.1s along with 802.1D. Those ports using 802.1s need to use a value other than 802.1s Disabled in its SID entry. Those ports using the 802.1s Disabled port state will use the port's Port State setting instead (i.e., 802.1D - Section 5.1.1).

The 802.1s Port State options take precedence over the port's PortState bits settings (Section 5.1.1), with the exception of the port's Disabled Port State (set in the port's PortState bits, Port offset 0x04). The port's Disabled Port State prevents all frames from entering and leaving the port so that has precedence over 802.1s Port States.

5.3 Special Frame Handling

The Special Handling block is used to modify the normal packet flow through the switch for special functions and/or to get specific frames to a VLAN isolated CPU. All ports have identical capabilities.

- Switching frames back out the port they came in on
- Tunneling frames through VLAN barriers based upon the frame's DA
- Mirroring ARP frames to the CPU through any VLAN barriers to the CPU
- Snoop (or trap) IGMP or MLD frames to the CPU through any VLAN barriers to the CPU

5.3.1 Switching Frames Back to their Source Port

The device supports the ability to send frames back out of the port on which they arrive. While this is not a standard way to handle Ethernet frames, some applications, like 802.3ah OAM loopback (Section 8.8.2), may require this ability on some ports. This feature can be enabled on a port-by-port basis by setting the port's own bit to a one in its VLANTable register (in the Port Based VLAN Map register, Port offset 0x06). This function is valid if 802.1Q is enabled on the port or not.

5.3.2 Tunneling Frames through VLANs

Normally frames cannot pass between port-based VLANs nor 802.1Q VLANs. The device can be configured to allow some frames to do so. Before a frame can tunnel through a VLAN barrier, its DA address must be loaded as static into the address database (see Section 9.1.1) and the VLANTunnel bit on the frame's Ingress port must be set to a one (see Port Control register, offset 0x04). When both of these conditions are true, the frame is sent out of the port or ports indicated in the static address's Destination Port Vector (the DPV field for the DA entry in the address database). The VLANTable (for In-chip Port Based VLANs), the cross-chip Port Base VLAN Table and 802.1Q membership data is ignored in this case. This feature is enabled only on those ports that have their VLANTunnel bit set to a one.

5.3.3 ARP Mirroring

The device supports ARP Mirroring on a per port basis. Mirroring is used to copy certain frames to the CPU for processing – tunneling them through VLAN barriers that may be in place to isolate the CPU from unnecessary frames. The required format of these frames is shown in Figure 24. The white portions of the frame in the figures are the only portions of the frame examined for this function.

Note

Two ARP frame formats are supported via a Global register setting:

- If ARPwoBC is zero (Global 1, offset 0x04), then ARPs must contain a Broadcast Destination address.
- If ARPwoBC is one, the ARPs only need an Ether type equal to 0x0806 and the frames Destination Address can be any value. This supports Mirroring ARP replies that are destined to a unicast address.

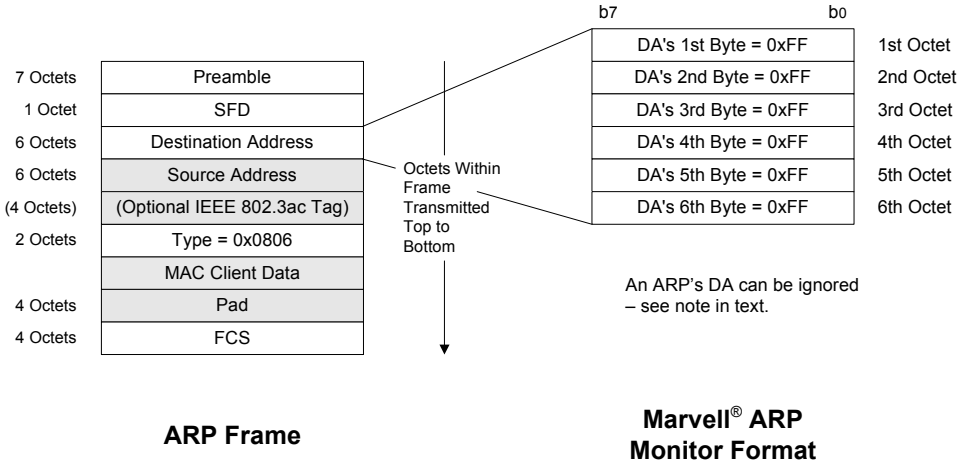
Management (MGMT) frames can still be ARP Mirrored (see Section 8.2). In this case the CPU will get two copies of the frame, one marked as a To_CPU BPDU (MGMT) and another marked as a To_CPU ARP Mirror, assuming the CPU's port is in DSA Tag mode (see Section 6).

When one of these frames enters a port where ARP Mirroring is enabled (by setting the port's ARP Mirror bit, Port offset 0x08), the frame is copied to the CPU's port as defined by the CPUDest[1] register (Global 1 offset 0x1A). The frame continues to be mapped to all the ports where it would have gone if ARP Mirroring was not enabled on the port. The frame will not get mirrored if it is filtered due to any rules enabled in the Ingress Policy block (Section 5). The CPU port (as defined by PortDest) does not have to be a member of the frame's VLAN to receive the frame. This is true for Port based VLANs (Section 5.2.1) and for 802.1Q based VLANs (Section 5.2.2).

Note

The queue priority (QPri) on ARP frames can be overridden. See Section 5.4.

Figure 24: ARP Mirror Format



1. ARP Mirrors always go out the port mapped by CPUDest. They do not go out the MirrorDest port which is reserved for Policy Mirrors (Section 5.1.3).

5.3.4 IGMP/MLD Trapping or Snooping

The device supports IPv4 IGMP snooping and IPv6 MLD snooping on a per port basis. Snooping is used to direct certain frames to the CPU for processing – tunneling them through VLAN barriers that may be in place to isolate the CPU from unnecessary frames. The required formats of these frames are shown in Figure 25 and Figure 26. The white portions of the frame in the figures are the only portions of the frame examined for this function.

Management (MGMT) frames bypass IGMP/MLD snooping (see Section 8.2).

When one of these frames enters a port where IGMP/MLD snooping is enabled (by setting the port's IGMP/MLD Snoop bit, Port offset 0x04), the frame is sent to the CPU's port as defined by the CPUDest register (Global 1 offset 0x1A) instead of where the frame normally would have been mapped. The frame will not get sent to the CPU if it is filtered due to any rules enabled in the Ingress Policy block (Section 5). The CPU port (as defined by PortDest) does not have to be a member of the frame's VLAN to receive the frame. This is true for Port based VLANs (Section 5.2.1) and for 802.1Q based VLANs (Section 5.2.2).

Note

The queue priority (QPri) on IGMP/MLD snooped frames can be overridden. See Section 5.4.

Figure 25: IPv4 IGMP Snoop Format

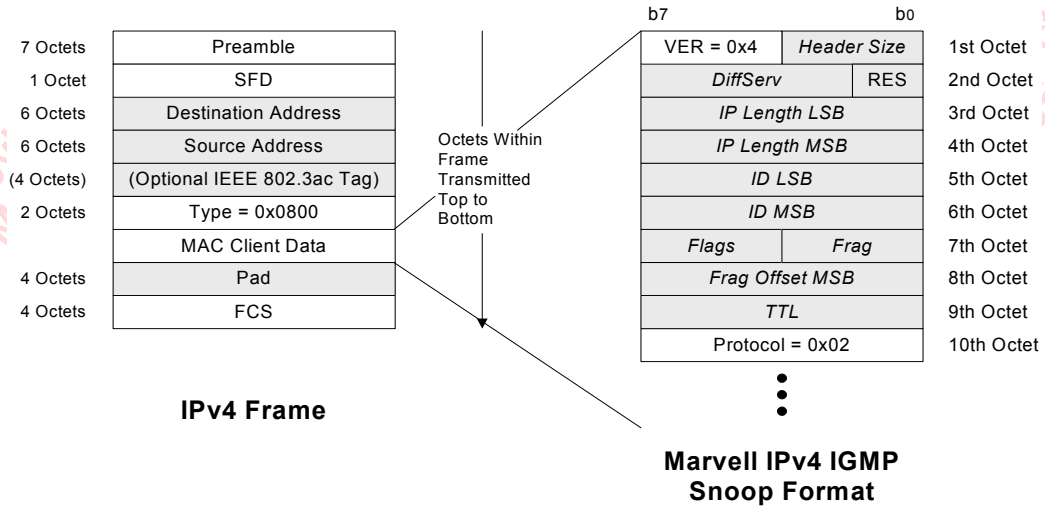
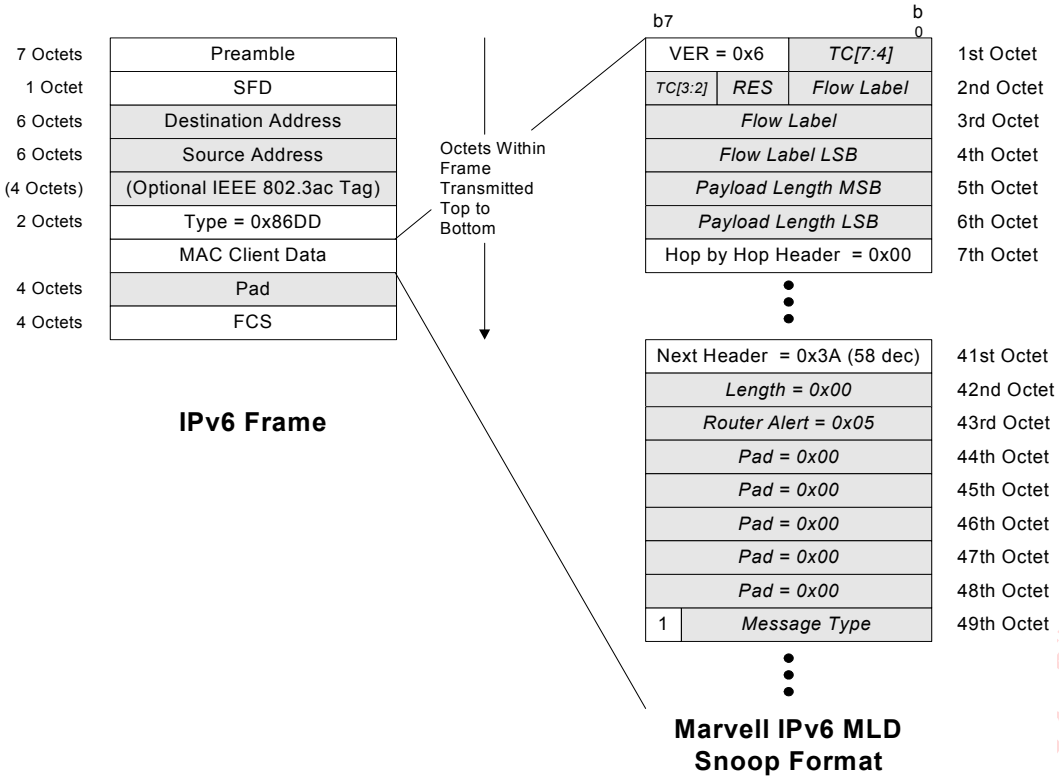


Figure 26: IPv6 MLD Snoop Format



5.4 Quality of Service (QoS) Classification

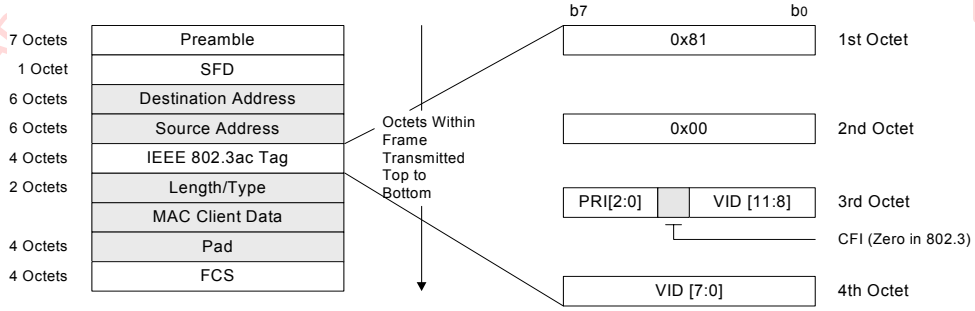
The Ingress block has the task of determining the priority of each frame to be used for the internal Queue Controller (QPri) as well as the priority assigned to the frame (FPri) if the frame egresses the switch tagged (Section 5.8.4). The Ingress block does not perform the QoS switching policy, which is the task of the Queue Controller (Section 5.6). Instead, it has the job of determining the QPri and FPri assigned to each frame for the Queue Controller and Egress block. The priority of a frame is determined by the following process (in this process order, therefore the last process step determines the final priority).

1. IEEE Tagged Frame Priority Extraction (Section 5.4.1)
2. IPv4 and IPv6 Frame Priority Extraction (Section 5.4.2)
3. Default Priority (Section 5.4.3)
4. Initial Priority Override (Section 5.4.4)
5. Frame Type Priority Override (Section 5.4.5)
6. Layer 2 Priority Override (Section 5.4.6)

5.4.1 IEEE Tagged Frame Priority Extraction

All ingressing frames with an 0x8100 ether type right after the frame's Source Address (see Figure 27) have their IEEE Tagged PRI bits mapped into the port's 8 entry x 3 bit IEEE Priority Remapping table (Figure 28 - port offsets 0x18 and 0x19). The result of this mapping is assigned as the frame's current FPri (the frame's final FPri value is remarked into the frame's IEEE Tagged PRI bits if the frame egresses a port tagged). This assignment occurs if 802.1Q is disabled or enabled on the port and occurs regardless of the frame's VID.

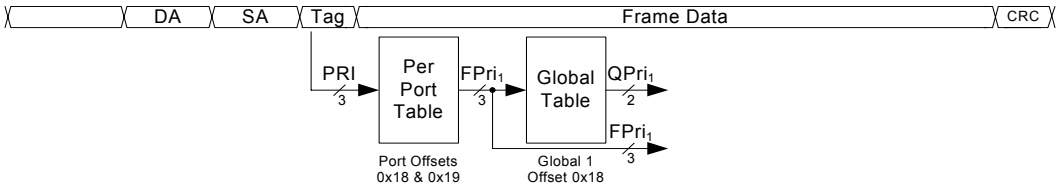
Figure 27: IEEE Tag Format



The IEEE Priority Remapping table can be used to scale some port's priorities down (for example 7:0 -> 3:0) while at the same time scaling some port's priorities up (for example 7:0 -> 7:4) or to ensure certain priorities are reserved for specific purposes by initially remapping all frames away from reserved priorities (for example 7:0 -> 4:0 protecting priorities 7:5). Later stage priority overrides can then be used to bring-up or restore a particular steam's FPri back to a higher level if needed.

The FPri (frame priority) is then mapped into the global IEEE PRI Mapping Table (Global 1, offset 0x18) in order to assign a QPri (queue priority) to the frame as well.

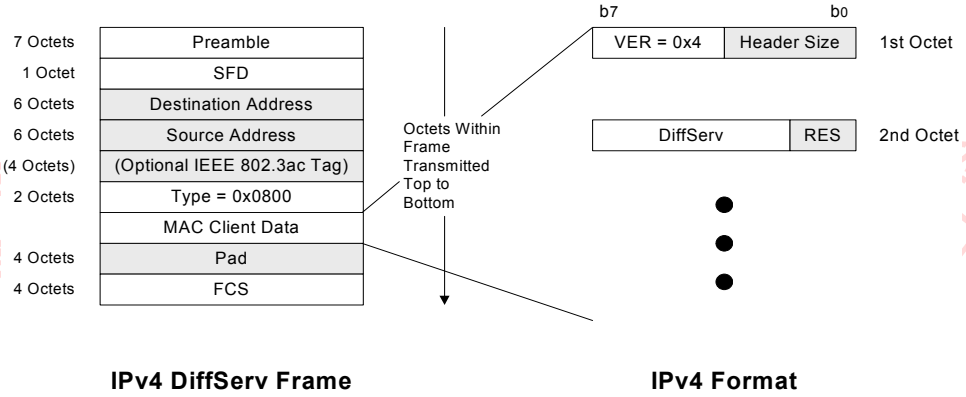
Figure 28: Port's IEEE PRI Mapping



5.4.2 IPv4 and IPv6 Frame Priority Extraction

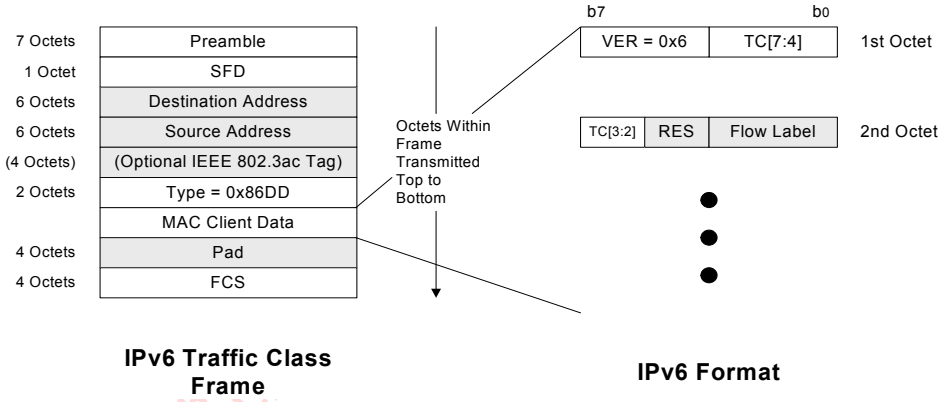
All ingressing frames with an 0x0800 ether type (IPv4) right after the frame's Source Address or right after an optional IEEE Tag (see Figure 29) have their VER bits checked. If the VER bits = 0x4 then the DiffServ bits are assigned as the frame's IP_PRI bits. These IP_PRI bits can be used to determine the frame's QPri and/or FPri bits (see Section 5.4.4).

Figure 29: IPv4 Priority Frame Format



All ingressing frames with an 0x86DD ether type (IPv6) right after the frame's Source Address or right after an optional IEEE Tag (see Figure 30) have their VER bits checked. If the VER bits = 0x6 then the upper 6 bits of the Traffic Class (TC[7:2]) are assigned as the frame's IP_PRI bits. These IP_PRI bits can be used to determine the frame's QPri and/or FPri bits (see Section 5.4.4).

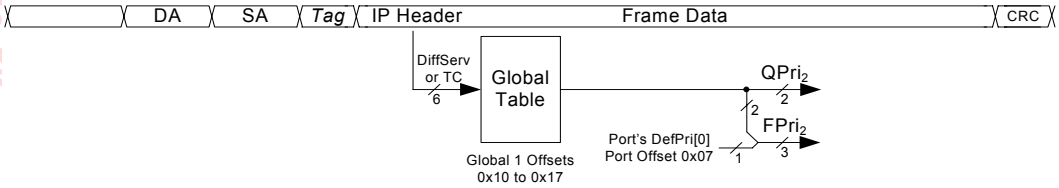
Figure 30: IPv6 Priority Frame Format



In both the IPv4 and IPv6 cases, the frame's IP_PRI bits are mapped into the global IP PRI Mapping Table (Global 1, offsets 0x10 to 0x17) in order to assign a QPri (queue priority) to the frame (Figure 31). The QPri bits are used as the upper two bits of the frame's FPri (frame priority). The least significant bit of the FPri in this case comes from the least significant bit of the port's DefPri (Default Priority, port offset 0x07).

If these frames egress out a port as IEEE tagged, the FPri value will be written to the frame's PRI bits (see Figure 27). This allows the IP priority to determine the tagged priority assigned to the frame.

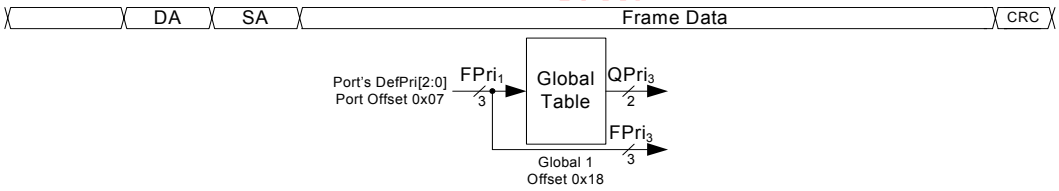
Figure 31: Port's IP Pri Mapping



5.4.3 Default Priority

All ingressing frames are assigned a default FPri (frame priority) and QPri (queue priority). The port's Default Priority (DefPri, port offset 0x07) is assigned as the frame's FPri. The FPri is then mapped into the global IEEE PRI Mapping Table (Global 1, offset 0x18) in order to assign a QPri to the frame (see Figure 32).

Figure 32: Port's Default PRI Mapping



5.4.4 Initial Priority Selection

Every frame entering the switch gets two priority values assigned to it. One priority value is used inside the switch only to determine which output queue the frame is to be mapped into. This is called QPri for queue priority. The other priority value is used outside the switch only to mark the frame's PRI bits if the frame egresses a port tagged (see Figure 27). This is called FPri for frame priority.

The QPri and FPri values are assigned differently to various frame types (see Section 5.4.1 to Section 5.4.3). Some frames are assigned more than one set of QPri and FPri values. For example an IEEE tagged IPv4 frame will get a set of values from the IEEE tag, another set from the IPv4 header and another set from the port. Which set should be used? Some applications require that the port's default priority be used regardless of the contents of the frame. How can this be accomplished? Table 31 shows how the port's InitialPri register bits along with the port's TagIfBoth register bit (both at Port offset 0x04) can be used to control which QPri and FPri values are assigned to a frame (prior to any priority overrides that may occur – Section 5.4.5 to Section 5.4.6).

Table 31: Initial QPri and FPri Selection

Selection	InitialPri	Ingressing Frame Type	TagIfBoth	Assigned Pri's	Figure and Meaning
Use port's defaults	0x0	All frame types	Don't Care	QPri = $QPri_3$ FPri = $FPri_3$	QPri = $QPri_3$/FPri = $FPri_3$ See Figure 32 - Use port's defaults
Support Tag priority only	0x1	Untagged frames	Don't Care	QPri = $QPri_3$ FPri = $FPri_3$	QPri = $QPri_3$/FPri = $FPri_3$ See Figure 32 - Use port's defaults
		Tagged frames		QPri = $QPri_1$ FPri = $FPri_1$	QPri = $QPri_1$/FPri = $FPri_1$ See Figure 28 - Use Tag's remapped priority
Support IPv4 or IPv6 priority only	0x2	Neither IPv4 nor IPv6 Frames	Don't Care	QPri = $QPri_3$ FPri = $FPri_3$	QPri = $QPri_3$/FPri = $FPri_3$ See Figure 32 - Use port's defaults
		IPv4 nor IPv6 Frames		QPri = $QPri_2$ FPri = $FPri_2$	Qpri = $QPri_2$/FPri = $FPri_2$ See Figure 31 - Use IP priority
Support IPv4, IPv6 and Tag priority	0x3	Neither IPv4 nor IPv6 nor tagged frames	Don't Care	QPri = $QPri_3$ FPri = $FPri_3$	QPri = $QPri_3$/FPri = $FPri_3$ See Figure 32 - Use port's defaults
		Untagged IPv4 or IPv6 frames		QPri = $QPri_2$ FPri = $FPri_2$	Qpri = $QPri_2$/FPri = $FPri_2$ See Figure 31 - Use IP priority
		Tagged frames that are not IPv4 nor IPv6		QPri = $QPri_1$ FPri = $FPri_1$	QPri = $QPri_1$/FPri = $FPri_1$ See Figure 28 - Use Tag's remapped priority
		Tagged IPv4 and Tagged IPv6 frames	0x0	QPri = $QPri_2$ FPri = $FPri_1$ (Special Case)	Qpri = $QPri_2$/FPri = $FPri_1$ See Figure 31 for IP QPri – Use IP QPri See Figure 28 for Tag FPri – Use Tag's remapped FPri priority
			0x1	QPri = $QPri_1$ FPri = $FPri_1$	QPri = $QPri_1$/FPri = $FPri_1$ See Figure 28 - Use Tag's remapped priority

Note

The default setting on the port's is to support IPv4, IPv6 and Tag priority (InitialPri = 0x3) and to choose a Tag's priority over the IPv4 or IPv6 priority (TagIfBoth = 0x1).

TagIfBoth being 0x0 controls only the selection of the internal QPri on frames that are both tagged and IP (with an InitialPri setting of 0x3). This allows the PRI bits in tagged frames to still come from the port's IEEE Priority

Remapping table while the internal QPri comes from the IP portion of the frame. If the PRI bits should be determined based on the IP priority of the frame instead, use an InitialPri setting of 0x2.

The InitialPri bits are enables on each of the priority classifications giving the designer any combination that is needed. For instance, if a port based higher priority port is required for a switch design the priority classification selections can be disabled by setting InitialPri to 0x0 on the port. This leaves the port's default priority resulting in all Ingressed frames being assigned the same priority on that port.

5.4.5 Frame Type Priority Override

After a frame's initial priority selection is made (Section 5.4.4) its queue priority (QPri) can be overridden either up or down based upon the frame's type. The frame's FPri cannot be modified in this way. This allows frames with a lower importance to be pushed to a lower priority inside the switch, while frames with a higher importance can be pushed to a higher priority.

The following frame types can have their QPri overridden in the following top to bottom processing order (as any one frame may meet more than one condition):

1. Broadcast – frame's DA = FF:FF:FF:FF:FF:FF
2. PolMirror – frame is being policy mirrored – Section 5.1.3
3. PolTrap – frame is being policy trapped – Section 5.1.3
4. EType – frame's ether type matches the port's PortEType register – Port offset 0x0F
5. ARP – frame is an ARP as determined by Section 5.3.3 even if ARP Mirror is disabled on the port (Port offset 0x07)
6. Snoop – frame is an IGMP/MLD as determined by Section 5.3.4 and if IGMP/MLD Snoop is enabled on the port (Port offset 0x04)

Each of these QPri overrides have an independent QPri value with an enable in the Priority Override Table (Global 2 offset 0x0F). If a frame is Broadcast, PolTrap, EType and ARP, the ARP entry will be accessed from the table (as it has the highest number in the list above) and used as the frame's QPri only if the ARP entry is enabled in the table. Only one access into the Priority Override Table is made per frame. So in this example, if the ARP entry is not enabled in the table then the frame will not get any QPri override even if one or more of the other possible frame types (Broadcast, PolTrap and EType in this example) had their QPri override enabled in the table. Because of this, it is best to fill in all entries in the table when the table is being used.

Notes

- The default settings in the table disable all frame type priority overrides.
- The Priority Override Table (Global 2 offset 0x0F) is also used for Secure Control on DSA ports – Section 7.8.

5.4.6 Layer 2 Priority Override

After a frame's initial priority selection is made (Section 5.4.4) and after any frame type priority override is carried out (Section 5.4.5) its queue priority (QPri) and/or its frame priority (FPri) can be overridden either up or down based upon the frame's DA, SA and/or VID. This allows frames with a lower importance to be pushed to a lower priority inside and/or outside the switch, while frames of with a higher importance can be pushed to a higher priority.

The following layer 2 fields can cause a frame's QPri and/or its FPri to be overridden in the following top to bottom processing order (as any one frame may meet more than one condition):

1. VID – the frame's VLAN ID (VID) is contained in the VTU with its UseVIDPri bits set to a one (Section 9.2.1) and if the port's VTU Pri Override bits are non-zero (Port offset 0x0D)
2. SA – the frame's Source Address (SA) is contained in the ATU with any of the many Priority Override Entry States (Section 9.1.1), and if the SA is assigned to the port, and if the port's SA Pri Override bits are non-zero (Port offset 0x0D)
3. DA – the frame's Destination Address (DA) is contained in the ATU with any of the many Priority Override Entry States (Section 9.1.1), and if the DA is static, and if the port's DA Pri Override bits are non-zero (Port offset 0x0D)

If a frame's VID is contained in the VTU with its UseVIDPri bit set (even if 802.1Q is disabled on the port), the frame's QPri will be overridden to the upper two bits of the VID entry's VIDPri bits if the port's VTU Pri Override register is set to a 0x2. Its FPri will be overridden to the three VIDPri bits if the port's VTU Pri Override register is set to a 0x1. If the port's VTU Pri Override register is set to a 0x3 then both the frame's QPri and FPri will be overridden as described above.

Then if a frame's SA is contained in the ATU with a Priority Override Entry State where the entry's DPV is associated with[1] the frame's source port, the frame's QPri will be overridden to the upper two bits of the SA entry's P bits if the port's SA Pri Override register is set to a 0x2. Its FPri will be overridden to the three P bits if the port's SA Pri Override register is set to a 0x1. If the port's SA Pri Override register is set to a 0x3 then both the frame's QPri and FPri will be overridden as described above.

Lastly, if a frame's DA is contained in the ATU with a Priority Override Entry State where the entry is static, the frame's QPri will be overridden to the upper two bits of the DA entry's P bits if the port's DA Pri Override register is set to a 0x2. Its FPri will be overridden to the three P bits if the port's DA Pri Override register is set to a 0x1. If the port's DA Pri Override register is set to a 0x3 then both the frame's QPri and FPri will be overridden as described above.

This order of processing places DA priority override above SA priority override which is above VID priority override (which is above frame type priority override – Section 5.4.5).

Note

The default settings disable all layer 2 priority overrides.

At this point the queue priority used inside the switch by the queue controller (QPri) and the frame priority that will be marked in the frame's IEEE Tag PRI bits if the frame egresses a port tagged (FPri), has been decided for the frame.

1. If the source port is a member of a Trunk (Section 8.10) the ATU Entry must match the source port's Trunk ID. If the source port is not a member of a trunk, the ATU Entry must have the source port's bit set to a one in its DPV.

5.5 Port Based Ingress Rate Limiting (PIRL)

The role of ingress rate limiting in switches has been increasing as more and more applications require accurate and predictable rate limiting of traffic entering any given port. The switches may need to either limit traffic at an aggregate level from a port or limit specific streams of traffic entering a port like unicasts, multicasts, unknowns etc. It may need to limit the rate for all frames but still keep QoS. The device supports this feature on a per rate resource basis.

The device supports 'Best-in-Class' per port TCP/IP ingress rate limiting along with independent Storm prevention. Port based ingress rate limiting accommodates information rates from 64 Kbps to 1 Mbps in increments of 64 Kbps, from 1 Mbps to 100 Mbps in increments of 1 Mbps and from 100 Mbps to 1000 Mbps in increments of 10 Mbps.

In addition to this, the device supports Priority based ingress rate limiting. A given ingress rate resource can be configured to track any of the four priority traffic types based on the frame's final QPri (refer to ingress policy section for priority classification details of incoming frames).

One of the popular schemes for implementing rate limiting is a leaky bucket. The way a leaky bucket scheme works is that the bucket drains tokens constantly at a rate called Committed Information Rate (CIR) and the bucket gets replenished with tokens whenever a frame is allowed to go through the bucket. All calculations for this bucket are done in tokens. Therefore, both bucket decrementing and incrementing is performed using tokens (i.e., frame bytes are converted into bucket tokens for calculation purposes).

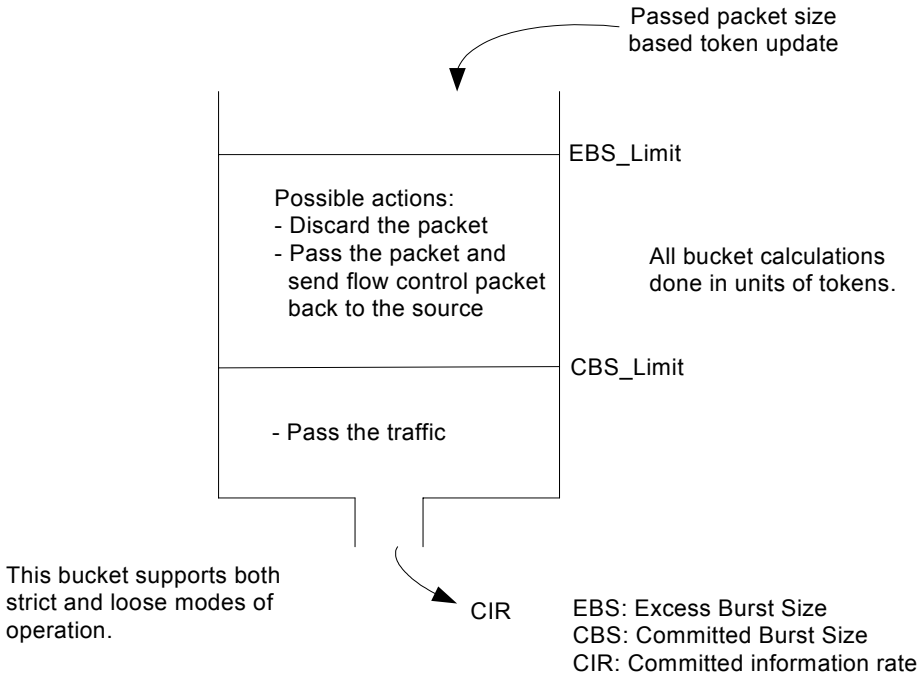
The device supports a color blind leaky bucket scheme. A color blind scheme implies that the frames are not expected to be marked prior to coming into the system. In some network elements it is required to have a color aware scheme of rate limiting where a frame marker would mark the frames to a lower priority than what they originally came in on if that traffic stream were to have violated any of the traffic rules. In the device's addressable applications, no such frame markers exist, thus a color blind scheme is employed.

As shown in Figure 33, the traffic below Committed Burst Size limit (CBS_Limit) is passed without any further actions. If the traffic burst were to continue and the bucket token depth approaches closer to the Excess Burst Size limit (EBS_Limit) by less than the CBS_Limit (i.e., EBS_Limit – CurrentBktDepth < CBS_Limit), then a programmable set of actions are specified.

Note

If the frame gets discarded then the equivalent number of tokens for that frame will not get added to the bucket.

Figure 33: Color blind Leaky Bucket for Rate Limiting



As shown in Figure 33, the traffic below CBS_Limit is passed without any further actions. If the traffic burst were to continue and the bucket token depth approaches closer to the EBSLimit by less than the CBSLimit (i.e., EBSLimit – CurrentBktDepth < CBSLimit), then a programmable set of actions are specified. Note that if the frame gets discarded then the equivalent number of tokens for that frame will not get added to the bucket.

If the EBS_Limit_action were programmed to be in flow control mode, then an Ethernet flow control frame gets generated and sent to the source port but the incoming frame gets passed through the rate resource. If the port is operating in half-duplex mode then the port gets jammed.

There are two rate limiting modes that are supported namely, "strict" and "loose" (strict and loose are traffic management terms).

In strict mode of operation, at any point in the rate resource, if there are not enough tokens to accept a complete frame the frame wouldn't get accepted i.e., there is no concept of negative tokens for a given rate resource. In loose mode of operation, if there are not enough tokens to accept a complete frame the frame would still get accepted by the rate resource and the token count might go above the EBSLimit but get clamped at the size of the bucket which is 2 **24-1. Note that in either strict or loose mode of operation when the token count exceeds the difference between EBSLimit and CBSLimit then the frame would not get accepted. The disadvantage of a strict implementation is that if there were to be a flow with Video/Audio streaming, as video frames are larger they may not get accepted but audio frames may go through. However, for TCP applications it is better to do a strict bucket implementation as large data frames won't get accepted till there is room for them and thus TCP ACK frames for previously transmitted frames could get accepted leading to lesser re-transmissions and better overall TCP throughput.

Another important feature that is supported by the ingress rate limiting block is Packet Sampling. In several network management applications, incoming frames from a given port may need to be sampled to a sampling port.

An example would be to sample 1 out of n frames from a given port. Any rate resource allocated for a given port can be configured to be in sampling mode. Once a rate resource is configured to be in sampling mode, it cannot be used to rate limit any packet types or priority traffic.

The ingress rate resources can be allocated for each port to cover various types of applications. TCP based applications could have the aggregate information flow for the TCP sessions running through that port limited. Other types include Broadcast storm prevention, TCP-SYN, TCP-FIN, MGMT, ARP attacks, priority aware rate limiting and packet sampling.

The following two diagrams indicate examples of how the five ingress rate resources that are available for each port on the 88E6097 device can be utilized.

Figure 34 describes a typical 88E6097 device Data Rate Limiting example. Figure 35 describes a typical 88E6097 device Priority Based Rate Limiting example.

Note

The 88E6096 device only supports the Bucket 0 and Bucket 1 Rate Limiting buckets.

Any given bucket can be programmed to be aggregate rate based or traffic type based.

Figure 34: Data Rate Limiting Example

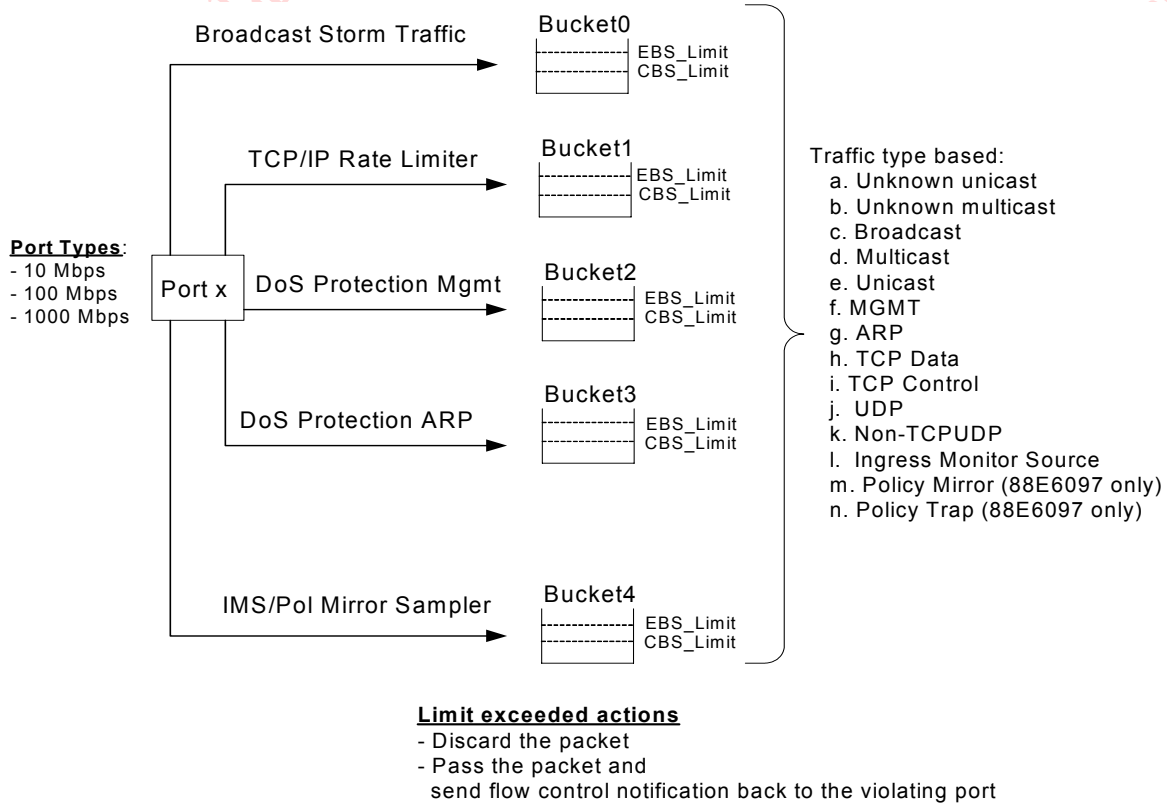
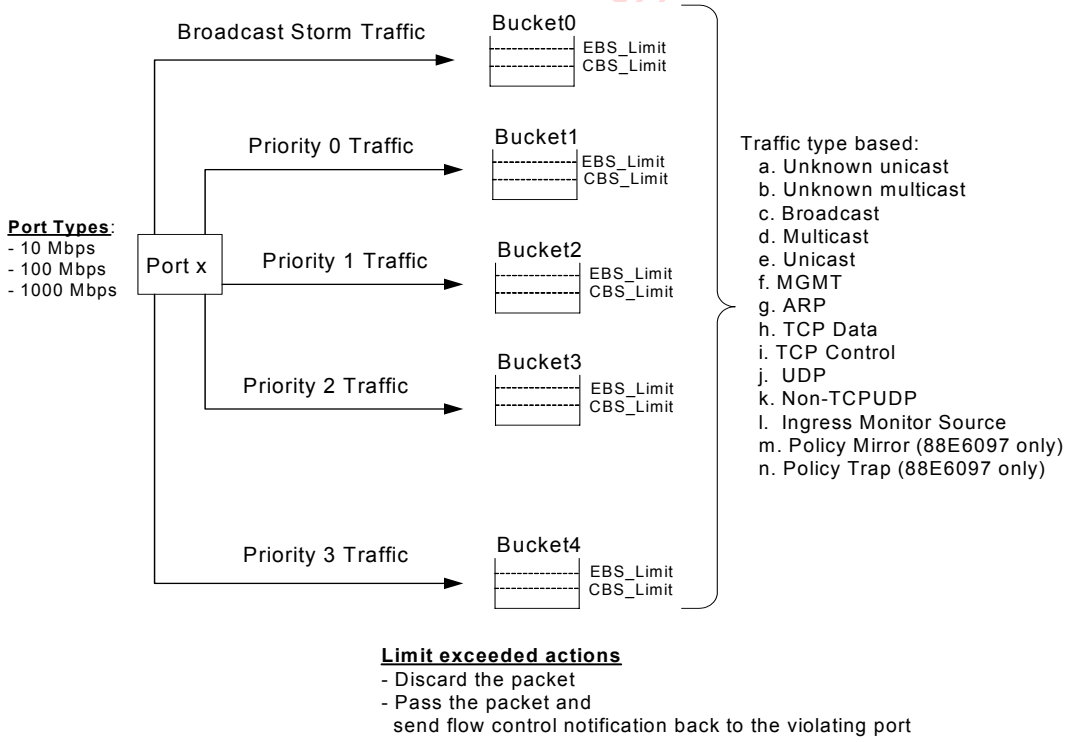


Figure 35: Priority Based Rate Limiting Example



Each port can be assigned up to five ingress rate resources.

Non-rate limiting (NRL) overrides can be programmed on a per-SA, per-DA or for frames that are classified as management frames by the ingress block.

The per-VID, per-SA or per-DA based NRL overrides are enabled by setting the corresponding bits in the register Ingress Rate Control register (Offset 0x09) bits 15 down to 13. Note that if either of the SA or DA non-rate limit's are set, then both ingress rate limiting logic would not account for that frame in their respective rate limiting calculations.

- Frame based
  - Count all Layer1 bytes
  - Count all Layer2 bytes
  - Count all Layer3 bytes

Where the definition of:

Layer1 = Preamble (8 bytes) + Frame's DA to CRC + IFG (12 bytes)

Layer2 = Frame's DA to CRC

Layer3 = Frame's DA to CRC - 18 - 4 (if frame tagged)

- TypeMask

  Any of the following frame types can be selected to be tracked as part of the rate resource calculations. As this is a bit vector, more than one frame type can be selected for a given rate resource.

  - Management (MGMT), Multicasts, Broadcasts, Unicasts, Address Resolution Protocol (ARP), TCP Data, TCP Ctrl, UDP, Non-TCPUDP, IMS, PolicyMorror, PolicyTrap, Unknown Unicasts or Unknown Multicasts
- RateType
  - Rate based or traffic type based
- Bucket_Increment (12 bits)[1]
- BucketRateFactor (16 bits)[1]
- EBS/CBS Limits (24 bits)[1]
- ActionMode
  - Discard
  - Send flow control back to the source port if flow control is enabled for that interface

  Flow control mode is expected to be programmed on ports that have a trusted flow control mechanism available. EBSLimitAction is a per-port characteristic. If a port has multiple rate resource buckets then all those buckets enabled are expected to be programmed with the same ActionMode.
- Sampling Mode

  This bit configures the rate resource to be in packet sampling mode. The devices support sampling of PolMirrored frames and for frames entering a port whose IMS bit (Switch Port register, offset 0x08) is set to 0x1. Note that frames that get discarded in any of the rate buckets for that port will not get accounted for in the sampling mode calculations. Also note that if the rate resource is configured to be in sampling mode operation, then the Bucket Rate Factor parameter described above is expected to be configured to 0x0 and the CountMode is expected to be configured to 0x0 i.e., Frame based.
- Account for Filtered discards

  This bit setting decides whether to account for frames that have been discarded because of other frame filtering reasons. To account for all frames coming into a given port(s) associated with this rate resource, this bit needs to be set.
- Account for Queue Congestion discards

  This bit setting decides whether to account for frames that have been discarded by the queue controller due to queue congestion reasons. To account for all frames coming into a given port(s) associated with this rate resource, this bit needs to be set.
- PIRLFCMode

  This bit determines when the EBSLimitAction is programmed to generate a flow control message, the deassertion of flow control is controlled by the PirlFCMode bit. When this bit is programmed to a zero, flow control gets de-asserted only when the ingress rate resource has become empty and when this is programmed to a one, flow control gets de-asserted when the ingress rate resource has enough room left as specified by the CBSLimit. For example if CBSLimit is programmed to 0x60000, then if there is room for at least 0x60000 tokens available in the rate resource bucket then a frame is accepted.
- PriOrPT (Priority or Packet Type)

  This defines whether the rate resource needs to derive the types of frames to track based on the priority of the incoming frame or based on the frame type.

1. Refer to the Bucket Configuration register (PIRL registers - Offset 0x00) for further details.

5.6 Queue Controller

The device's queue controller uses an advanced non-blocking, four priority, output port queue architecture with Resource Reservation. As a result, the device supports definable frame latencies with guaranteed frame delivery (for high priority frames) without head-of-line blocking problems or non-blocked flow disturbances in any congested environment (for all frame priorities).

5.6.1 Frame Latencies

The device can guarantee frame latencies owing to its unique, high performance, four priority queuing system. A higher priority frame is always the next frame to egress a port when a port is currently egressing frames of a lower priority[1]. This is true regardless of the priorities and regardless the egress port's Scheduling[2] mode of the switch.

5.6.2 No Head-of-Line Blocking

An output port that is slow or congested never effects the transmission of frames to uncongested ports. The device is designed to ensure that all uncongested flows traverse the switch without degradation regardless of the congestion elsewhere in the switch.

5.6.3 QoS with and without Flow Control

The Queue Controller is optimized for three modes of operation:

- Flow Control disabled on all ports
- Flow Control enabled on all ports
- Flow Control enabled on some ports and disabled on the rest

When flow control is enabled no frames are dropped and higher priority flows receive higher bandwidth through the switch (i.e., these flows are less constrained if there is congestion between a higher and a lower priority flow). The percentage of bandwidth that each flow receives is determined by the Scheduling mode see (Section 5.6.6). Flow control prevents frames from being dropped but it can greatly impact the available bandwidth on any network segment that is utilizing flow control. The latency of higher priority data on flowed-off network segments also increases since industry constrained standard flow control mechanisms stop all frames from being transmitted. Therefore, flow control may not be desirable in a QoS switch environment. For this reason, the device is also optimized to work properly without flow control.

When flow control is disabled and congestion occurs for an extended period of time, frames will be dropped. This is true in all switches. The device drops the correct frames, i.e., the lower priority frames. In this case, the higher priority flows get a higher percentage of the free buffers. The percentage of buffers they get is determined by the Scheduling mode (see Section 5.6.6).

For the mixed flow control case, frames are dropped if congestion occurs on the non-flow controlled paths while the flow controlled paths do not drop any frames. The percentage of bandwidth each port receives is priority based and fairness is maintained between all the ports in the switch (determined by the Scheduling mode). In the mixed mode case, frames will not be dropped only if both the source and the destination port(s) of the frame have flow control enabled. If a flow control enabled ingress port maps a frame to a non-flow control enabled egress port the frames will be dropped if congestion occurs (the same drop action will occur if the ingress port is not flow control enabled while the egress port has flow control enabled).

1. This is true as long as the default Weighting Table is used - Section 5.6.7.
2. The Scheduling mode selects a strict priority, an 8-4-2-1 weighted round robin, or a mixture of the two – Section 5.6.7

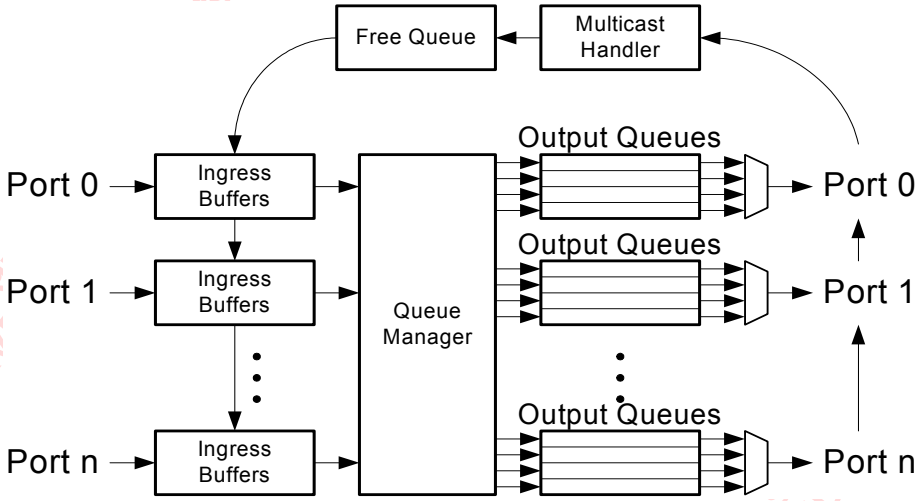
5.6.4 Guaranteed Frame Delivery without Flow Control

The device can guarantee high priority frame delivery[1], even with Flow Control disabled, due to its intelligent Resource Reservation system. Having an output queue with multiple priorities is not sufficient to support Quality of Service (QoS) if the higher priority frames cannot enter the switch due to a lack of buffers. The device reserves buffers for higher priority frames so they can be received and then switched. These high priority buffers are the first to get replenished from the Free Queue which prepares the receiving port for the next high priority frame.

5.6.5 The Queues

The queues in the device are shown in Figure 36.

Figure 36: Switch Queues



5.6.5.1 Queue Manager

At reset[2], the Queue Manager initializes the Free Queue by placing all the buffer pointers into it and ensures that all the other queues are empty. Then it takes the first available free buffer pointers from the Free Queue and assigns them to any Ingress port that is not Disabled[3] and whose link[4] is up. The switch is now ready to accept and switch packets. Whenever any port's Link goes down or if the port is set to the Disabled Port State the port's ingress Buffers and Output Queue buffers are immediately returned to the Free Queue. This prevents stale or lost buffers and allows the Free Queue to be as large as possible so larger bursts of momentary congestion can be handled. When a non-Disabled port's Link comes back up it gets its Ingress Buffers back so it can start receiving frames again.

When a MAC receives a packet it places it into the embedded memory at the address pointed to by the Input Pointers it received from the Queue Manager. When the packet is received, the MAC transfers the pointer(s) to the Queue Manager and requests new buffers from the Free Queue. If the Free Queue is empty the MAC does not

1. If all the frames entering a port are all high priority at wire speed their delivery cannot be guaranteed.
2. The Queue Manager is reset by either the hardware RESETn pin or a software reset by the SWReset bit in the Switch Global Control register (Global 1 offset 0x04).
3. If a port is in the Disabled Port State (Port offset 0x04) its Ingress buffers are left in the Free Queue for other ports to use.
4. SERDES-based ports need to get a Link up from the PHY, (G)MII based ports have Link up if they are enabled.

receive any pointers until they become available. If the MAC starts to receive a packet when it has no pointers, the packet will be dropped. Flow control will be asserted before this occurs if it's enabled.

The Queue Manager uses the data returned from the Lookup Engine (see Section 4.4.1) and the Ingress Logic (see Section 5 to Section 5.4) to determine which Output Queue or Queues the packet's pointer should go to and at what priority. At this point, the Queue Manager modifies the desired mapping of the frame depending upon the mode of the switch and its level of congestion. Two modes are supported, with and without Flow Control. Both modes are handled at the same time and can be different per port (i.e., one port has Flow Control enabled and another has it disabled).

If Flow Control is enabled on an Ingress port the frame is switched to the desired Output Queues if the egress port also has Flow Control enabled. This is done so frames are not dropped. Instead, the Queue Manager carefully monitors which Output Queues are congested and enables or disabled Flow Control on the Ingress ports that are causing the congestion. This approach allows uncongested flows to progress through the switch without degradation.

If Flow Control is disabled on an Ingress port or the Egress port the frame may be discarded instead of being switched to the desired Output Queue(s). If a frame is destined to more than one Output Queue it may get switched to some and not others. The decisions are complex as the Queue Manager takes many pieces of information into account before the decision is made. The Queue Manager looks at the priority of the current frame, the current level of congestion in the Output Queue(s) the frame is being switched to and the current number of free buffers in the Free Queue. The result is uncongested flows transverse the switch unimpeded and higher priority frames get in and through the switch faster even if there is congestion elsewhere in the switch.

5.6.5.2 Output Queues

The Output Queues receive and transmit packets in the order received for any given priority. This is very important for some forms of Ethernet traffic. The Output Queues will be emptied as fast as they can – but they could empty at different rates. This could be due to a port being configured for a slower speed or it could be caused by network congestion (collisions or flow control).

Each port contains four independent Output Queues, one for each priority. The order the frames are transmitted out each port is controlled by the port's Scheduling bits (Port offset 0x0A). Strict, weighted round robin priority, or a mixture of the two can be selected (see Section 5.6.6). The weighted round robin is programmable (Section 5.6.7).

After a packet has been transmitted fully out to the MAC, the Output Queue passes the transmitted packet's pointer(s) to the Multicast Handler for processing. The MAC then begins transmitting the next packet.

5.6.5.3 Multicast Handler

The Multicast Handler receives the pointers from all the packets that were transmitted. It looks up each pointer to see if this packet were directed to more than one Output Queue. If not, the pointers are returned to the Free Queue where they can be used again. If the frame was switched to multiple Output Queues the Multicast Handler ensures that the frame has egressed all of the ports to which it was switched before returning the pointer(s) to the Free Queue.

5.6.6 Per Port Fixed or Weighted Priority

The device supports strict priority, weighted round robin, or a mixture on a per egress port selection basis. The selection is made by the Scheduling bits at Port offset 0x0A. In the strict priority scheme all top priority frames egress for a port until that priority's queue is empty, then the next lower priority queue's frames egress, etc. This approach can cause the lower priorities to be starved out preventing them from transmitting any frames but also ensures that all high priority frames Egress the switch as soon as possible. In the weighted scheme an 8, 4, 2, 1 weighting is applied to the four priorities unless an alternate weighting is programmed into the QoS Weights Table

(Section 5.6.7). This approach prevents the lower priority frames from being starved out with only a slight delay to the higher priority frames.

Some applications may require the top priority queue, or the top two priority queues to be in a fixed priority mode while the lower queues work in the weighted approach. All scheduling modes are selectable on a per port basis using the port's Scheduling bit at Port offset 0x0A as follows:

- 0x0 = Use weighted round robin for all queues
- 0x1 = Use strict priority for queue 3 and weighted round robin for queues 2, 1 and 0
- 0x2 = Use strict priority for queues 3 and 2, and weighted round robin for queues 1 and 0
- 0x3 = Use strict priority for all queues

5.6.7 Programmable Weighting Table (88E6097 Only)

The device supports a programmable weighting table that is global for all ports. The table not only defines the weights of each priority queue, but also the sequence order as to how frames are to egress the ports. Any sequence and weighting can be defined that can fit into 128 sequence steps.

5.6.7.1 The Default Table

The default table uses an 8, 4, 2, 1 weighting meaning 8 entries for priority 3, 4 entries for priority 2, etc. In sequence order, this weighting table looks like: 3, 2, 3, 1, 3, 2, 3, 0, 3, 2, 3, 1, 3, 2, 3 where the numbers 0 to 3 represent priorities 0 to 3 respectively. Counting each of the numbers in the sequence shows there are 15 steps (which is also equal to the sum of the weights 8+4+2+1=15).

The ordering of this table defines the priority order in which frames will egress a port assuming all four priority queues on the port are full. The sequence is important because it defines when higher priority frames are allowed to egress the port. Look at the default sequence in the paragraph above. If priority queue 0 is the only queue with frames in it, these frames will egress the port at full wire speed. This is because the 'next queue to service' decision is always accomplished in one clock. So if the 'queue to service' pointer is in the middle of the list at '0', and if queues 3, 2, and 1 are empty, then the pointer will jump back around to '0' in one clock and then next frame to egress the port will come from queue 0.

While this is occurring, if a new frame is mapped into the port's egress queue at priority 1 it will be the next frame to egress because the number '1' occurs in the sequence list before then next '0' does. The same thing will occur if the new frame was mapped at priority '2' or '3'.

Suppose queue 2 is the only queue with frames in it. These frames will egress the port at full wire speed as the 'queue to service' pointer goes from the 1st '2' in the list to the next one and so on. If queue 1 gets some frames in its queue the sequence would be: 2, 1, 2, 2, 1, 2 which repeats as 1, 2, 2, 1, 2, 2,... as the '3's and '0's are skipped. More importantly, if a priority 3 frame comes along in this example, it will always be the next frame to egress since a '3' follows every other entry in the list. So no matter where the current 'queue to service' pointer is in the list, queue 3 will always be next. The same statement can be made for priority 2 if queue 3 is empty. The number '2' follows every '1' and '0' assuming '3' is skipped (because it was empty).

5.6.7.2 Alternate Weighting Tables

The default 8, 4, 2, 1 weighting sequence is:

3, 2, 3, 1, 3, 2, 3, 0, 3, 2, 3, 1, 3, 2, 3 (a 15 step sequence)

The same 8, 4, 2, 1 weighting can be programmed into the device as the following sequence:

3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1, 0 (a 15 step sequence)

While this sequence has the same weights, it ordering is very different! While this sequence allows larger bursts from the higher priorities, it does so at a latency penalty. For example, assume the current 'queue to service'

pointer is pointing to the 1st '2' in the list when a priority '3' frame comes along. The priority 3 frame may have to wait behind six frames before it can egress (three more '2's, two '1's and one '0'). Keep this in mind when defining an alternate weighting.

To support bursts of priority '3' frames while at the same time ensuring they are always the next frame out of the switch, the following weighting sequence could be used:

3, 3, 3, 3, 2, 3, 3, 3, 3, 1, 3, 3, 3, 3, 2, 3, 3, 3, 3, 0, 3, 3, 3, 3, 2, 3, 3, 3, 3, 1, 3, 3, 3, 3, 2, 3, 3, 3, 3

(a 24, 4, 2, 1 weighting sequence with 31 steps)

5.6.7.3 Programming the QoS Weighting Table

Program the QoS Weighting Table as follows. The default 3, 2, 3, 1, 3, 2, 3, 0, 3, 2, 3, 1, 3, 2, 3 weighting sequence will be used as an example:

1. Define your weighting sequence. Make sure all queues (3, 2, 1 and 0) show up at least once in the sequence. If a queue is left out of the sequence frames will never egress from that queue and the memory buffers used by these frames will be forever stuck behind the port.
2. Take the 1st four entries in the sequence (3, 2, 3, 1) and reverse their order (to 1, 3, 2, 3).
3. Write each sequence number in binary in the new order to form an 8-bit value (0111 1011).
4. Write this 8-bit value to the 1st pointer (0x00) in the QoS Weights register (Global 2, offset 0x1C).
5. Repeat for the next 4 entries in the sequence until there are no more entries. Assume zeros for non-existent entries. Loading the example is continued as follows:
   a) 3, 2, 3, 0 becomes 0011 1011 written to pointer 0x01
   b) The second 3, 2, 3, 1 becomes 0111 1011 written to pointer 0x02
   c) The last 3, 2, 3 becomes 0011 1011 written to pointer 0x03 – the first 00 is the non-existent entry
6. Unused entries do not need to be written
7. Write the sequence length (0x0F since there are 15 steps in the example) to pointer 0x20. This defines the length of the sequence and causes the new sequence to be installed and used (prior to performing this write the previous QoS weight table is used).

Note

All QoS weight values (3, 2, 1 and 0) must show up in the table at least once or improper switch operation will occur.

5.7 Embedded Memory

The device's MACs interface directly to the embedded 1Mb (4Kx256) Multi-port Synchronous SRAM (MP-SSRAM). The SSRAM is running at 125 MHz and the data bus is 256 bits wide. The memory interface provides up to 32 Gigabits per second bandwidth for packet reception/transmission. This memory bandwidth is enough for all of the ports running at full wire speed in full-duplex mode with minimum size frames.

5.8 Egress Policy

The Egress Policy block is used to modify the normal packet flow through the switch, by limiting which frames are allowed to egress, as well as modifying or updating the contents of the frames as they egress. All ports have identical capabilities.

- Non-management frame types can be blocked from leaving the switch to support Spanning Tree Protocol (Section 5.1.1 for classic 802.1D and Section 5.2.3 for 802.1s).
- Frames with an unknown unicast Destination Address (i.e., one not found in the address database – Section 4.4.1) can be prevented from egressing any port (Section 5.8.1).
- Frames with an unknown multicast Destination Address (i.e., one not found in the address database – Section 4.4.1) can be prevented from egressing any port (Section 5.8.2).
- 802.1Q Secure and Check Mode can filter out frames whose VID is not contained in the VLAN database (Section 5.8.3).
- Port based VLANs using the port's EgressMode bits (Port offset 0x04) and/or 802.1Q VLANs using the VID's MemberTag bits as stored in the VTU (Section 9.2.1) are used to determine if a frame is to be transmitted Unmodified, Tagged or UnTagged (Section 5.8.4).
- Port based Egress Rate Shaping is supported with Frames-per-Second or Layer 1, Layer 2 or Layer 3 Bits-per-Second with support for 802.3ar's Frame Overhead (Section 5.8.5).

Note

Provider Tagging, otherwise known as Double Tagging or Q-in-Q Tagging is also supported. See Section 6.

5.8.1 Forward Unknown/Secure Port

The device can be configured to prevent the forwarding of unicast frames with an unknown destination address (i.e., the address is not present in the address database – Section 4.4.1). The forwarding can be prevented on a per port basis by adjusting the port's EgressFloods bits in the Port Control register (Port offset 0x04) so that frames with unknown unicast addresses only go out from the port or ports where a server or router is connected.

This, together with the disabling of automatic address learning on a port (Section 4.4.3), allows a port to be configured as a Secure Port. A Secure Port allows communications to and from approved devices (by MAC address) only. In this mode all approved devices need to have their MAC address loaded as static into the address database (Section 9.1.1). When a new device tries to access the network through a Secure Port that new device's address is not automatically learned (learning must be disabled on Secure Ports) but its frame can progress to the server. When the server replies by sending a unicast frame back to the new device's address, that frame does not make it to the new device since the new device's address is not in the address database and frames with unknown unicast addresses are not allowed to egress the Secure Port. This effectively ends the communication between the un-approved new device and the rest of the network. Secure Port is similar to the 802.1X (Section 4.4.7) without the authentication server and the associated interrupts to the CPU.

5.8.2 Forward Unknown for Multicasts

The device can be configured to prevent the forwarding of multicast frames with an unknown destination address (i.e., the address is not present in the address database – Section 4.4.1). The forwarding can be prevented on a per port basis by adjusting the port's EgressFloods bits in the Port Control register (Port offset 0x04) so that frames with unknown multicast addresses only go out from the port or ports where a server or router is connected.

By default, the Broadcast address (FF:FF:FF:FF:FF:FF) is considered a multicast address and frames with this destination address will not egress ports configured to prevent unknown multicasts from egressing. Two other options for the Broadcast address are available:

1. Load the Broadcast address into the address database so that it is known. This address would need to be loaded into each FID (Forwarding Information Database - Section 4.4.8) that requires Broadcast frame's to egress – giving control over which FIDs allow Broadcasts.
2. Or, consider frames with the Broadcast address to be special and not part of the group of other unknown multicast addresses, which is accomplished by setting the global FloodBC bit to a one (Global 2, offset 0x05). This allows frames with the Broadcast destination address to egress all ports even if the port's EgressFloods bits prevent unknown multicasts from egressing.

Note

A port's EgressFloods bits prevent frames from egressing the port, only if the frames would have egressed the port when EgressFloods is in its default setting (i.e., allow all unknown frames to egress).

5.8.3 Secure 802.1Q VLANs

Egress security filtering can be performed on any egress port whose 802.1Q Mode (Port offset 0x08) is Secure or Check. This feature is useful when a frame enters the device on a port where the 802.1Q Mode is Fallback or 802.1Q Disabled. In this case the frame may get mapped to a port configured in 802.1Q Secure or Check Mode even if the VID assigned to the frame during ingress is not in the VLAN database (i.e., the frame is mapped using the ingress port's Port Based VLANs). The default action of 802.1Q Mode Secure or Check is to discard the egressing frame since its VID is not in the VTU.

This egress security filtering policy can be disabled by setting the global NoEgrPolicy bit to a 1 (Global 2, offset 0x1D).

Note

If a Tagged frame enters a port where the 802.1Q Mode is Disabled, the VID used for the egress Secure or Check functions is the source port's DefaultVID (Port offset 0x07).

5.8.4 Tagging and Untagging Frames

Egress tagging and untagging is supported dynamically using 802.1Q VLANs, or statically using Port Based VLANs. The mode that is used on a port is determined by the egress port's 802.1Q Mode bits (Port offset 0x08) as follows:

- Secure or Check – The MemberTag bits (in VTU Data register, Global 1 offsets 0x07 to 0x09) associated with the VID assigned to the frame during ingress determines if the frame should egress unmodified, tagged or untagged (assuming egress security filtering is being performed – Section 5.8.3, else Secure and Check works like Fallback below).
- Fallback – The MemberTag bits associated with the VID assigned to the frame during ingress determine if the frame should egress unmodified, tagged or untagged if the VID is contained in the VLAN database (Section 9.2.1). If the VID is not found in the VLAN database, or if the VID is found with the port's MemberTag bits set to 0x3, the frame egresses unmodified, tagged or untagged determined by the port's EgressMode bits (Port offset 0x04).
- 802.1Q Disabled – The port's EgressMode bits determine if the frame will egress unmodified, tagged or untagged.

The device performs the following actions on the egressing frame depending upon the Egress tagging mode that was determined above:

- Transmit Unmodified[1] - UnTagged frames egress the port UnTagged. Tagged frames egress the port Tagged.
- Transmit UnTagged[2] - UnTagged frames egress the port unmodified. The IEEE Tag on Tagged frames is removed, the frame is zero padded if needed[3], and a new CRC is computed for the frame.
- Transmit Tagged[4] - Tagged frames egress the port unmodified. An IEEE Tag is added to UnTagged frames and a new CRC is computed. The contents of the added tag is covered in Section 5.8.4.1.

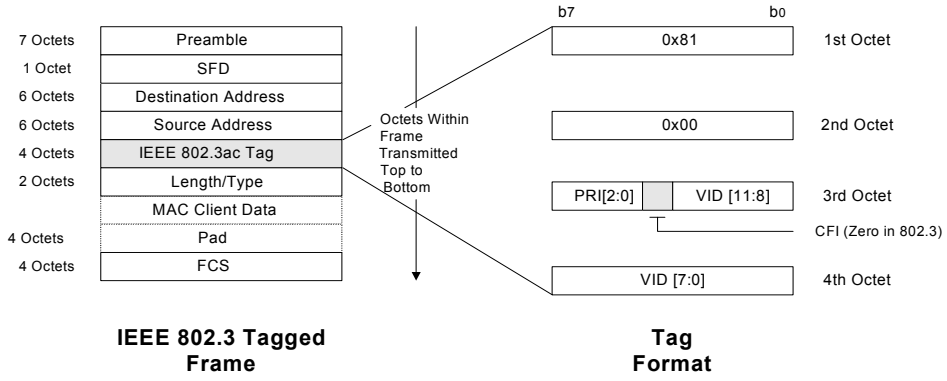
The format of an IEEE Tagged frame is shown in Figure 37.

Notes

- A physically Tagged frame is considered Tagged for the above egress frame modifications only if it enters a port where 802.1Q is enabled (i.e., the port's 802.1Q Mode in Port offset 0x08 is Secure, Check or Fallback). If a switch is configured where some ports have 802.1Q enabled (using Q VLANs) and some ports have 802.1Q disabled (using Port Based VLANs) extreme care is required to prevent physically Tagged frames from egressing double tagged (because physically Tagged frames are considered UnTagged for egress frame modifications if they enter a port where 802.1Q is Disabled). The best way to prevent this is set all port's EgressMode bits (Port offset 0x04) to Transmit Unmodified. This ensures frames are transmitted looking the way they did when they entered the switch unless the egress port's 802.1Q is Enabled and the frame's VID is in the VTU. Make sure the DefaultVID (Port offset 0x07) on ports where 802.1Q is Disabled is not contained in the VTU, or if it is, make sure its MemberTag bits are set to Transmit Unmodified.
- If a Tagged frame is Transmitted Tagged, its VID is updated to be the VID assigned during Ingress (Section 5.2.2.5) and its PRI bits are updated to be the FPri assigned during Ingress (Section 5.4).

1. This is the default setting so the switch acts as a transparent switch.
2. Needed when switching frames to end stations that don't understand Tags.
3. Tagged frames that are less than 68 bytes are padded with zero data to ensure the UnTagged frame is at least 64 bytes in size.
4. Typically used when switching frames into the core or up to a server.

Figure 37: IEEE Tag Format



5.8.4.1 Adding a Tag to Untagged Frames

When a Tag is added to an UnTagged frame the Tag is inserted right after the frame's Source Address. The four bytes of added data are:

- The 1st Octet is always 0x81
- The 2nd Octet is always 0x00
- The PRI bits indicate the frame's priority (FPri) determined during Ingress (Section 5.4)
- The CFI bit is always set to a zero
- The VID bits indicate the VID assigned to the frame during Ingress (Section 5.2.2.5)

Note

A Tag that is added due to Provider Tagging is done differently. See Section 6.

5.8.4.2 Priority Re-Map and Priority Override

When a Tagged frame egresses a port Tagged, the PRI bits in the tag are modified to reflect the frame's priority (FPri) that was determined during Ingress (Section 5.4). The PRI bits can be re-mapped by the ingress port's IEEE Priority Remapping registers (Port offsets 0x18 and 0x19) or the frame's PRI bits can be ignored and the ingress port's default priority used instead, or the frame's priority can be overridden.

Note

A physically Tagged frame is considered Tagged for egress frame modification purposes only if the frame entered a port whose 802.1Q Mode is enabled (Port offset 0x08).

5.8.4.3 VID 0x000 and VID Override

A Tagged frame egressing a port Tagged may have its VID bits modified. If the Ingressing frame's VID was 0x000, or if the Ingress port's DefaultVID (Port offset 0x07) is assigned to the frame instead. See Section 5.2.2.5.

Note

A physically Tagged frame is considered Tagged for egress frame modification purposes only if the frame entered a port whose 802.1Q Mode is enabled (Port offset 0x08).

5.8.5 Egress Rate Shaping

A switch design may need to limit the transmission rate of egressing frames but still keep QoS. The device supports this on a per-port basis by setting bits in the port's Egress Rate Control registers (Port offsets 0x09 and 0x0A). Egress rate limiting is performed by shaping the output load on a per-frame basis (i.e., the inter frame gap is increased between each transmitted frame to meet the selected mode).

The supported modes are:

- Count frames for a Frames-per-Second rate
- Count Layer 1, Layer 2 or Layer 3 bytes for a Bits per Second rate adding an optional fixed Frame Overhead adjustment per frame to account for upstream frame modifications

5.8.5.1 Frames-per-Second Egress Shaping

Each port can count frames for a Frames-per-Second rate in the range of:

- 7700 frames per second up to 1,490,000 frames per second.

Note

For reference: 1518 bytes frames at 100 Mbps is 8128 frames per second and 64 byte frames at 1000 Mbps is 1,488,095 frames per second.

Two registers are used to set the desired Frames-per-Second rate. These are EgressRate (Port offset 0x0A) and EgressDec (Port offset 0x09). The formula to use is:

EgressRate = EgressDec / (32 ns * Desired Egress Frame Rate per Second)

Note

It is recommended that EgressDec = 0x01 when counting Frames-per-Second

For Example: Set the egress rate to 7700 frames-per-second

EgressRate = EgressDec / (32 ns * 7700 frames-per-second)

EgressRate = 1 / (0.000000032 * 7700)

EgressRate = 4058.44 or 0xFDA

EgressDec = 0x01

Due to the cropping of fractions, the actual rate will be:

ActualRate = EgressDec / (32 ns * EgressRate)

ActualRate = 1 / (0.000000032 * 4058)

ActualRate = 7700.84 frames per second

Note

If the desired rate is a 'not to exceed' rate then increment the EgressRate register value by 1 if there is a non-zero fraction in the decimal result. In the above example a value of 0xFDB for would be needed for a 'not to exceed' rate of 7700 frames-per-second. In this case, the ActualRate will be 7698.94 FPS.

5.8.5.2 Bits-per-Second Egress Shaping

Each port can count bytes for a Bits-per-Second rate in the range of:

- 64 kbps to 1 Mbps in 64 kbps steps
- 1 Mbps to 100 Mbps in 1 Mbps steps
- 100 Mbps to 1000 Mbps in 10 Mbps steps

Note

Other rates in between those stated above are possible by using the generic equation stated below, but only the rates stated above have been verified.

Three registers are used to set the desired Bits-per-Second rate. These are CountMode and EgressRate (both at Port offset 0x0A), and EgressDec (Port offset 0x09).

CountMode is used to select which bytes in the frames to count as follows:

- Count Layer 1 bytes (8 Preamble bytes + Frame's DA to CRC + 12 IFG bytes)
- Count Layer 2 bytes (Frame's DA to CRC)
- Count Layer 3 bytes (Frame's DA to CRC – 18 Layer 2 bytes – 4 byte if frame is Tagged)

EgressRate and EgressDec are used to set the desired Bits-per-Second using the following formula:

EgressRate = 8 bits * EgressDec / (32ns * Desired Egress Bit Rate per Second)

Notes

- When egress rate shaping in the 64 kbps to 1 Mbps range it is recommended to set EgressRate = 0xF42 (to select 64 kbps steps) and use EgressDec to select the number of 64 kbps steps to use. For example: set EgressDec to 0x01 for 64 kbps, set it to 0x02 for 128 kbps, set it to 0x03 for 192 kbps, etc.
- When egress rate shaping in the 1 Mbps to 100 Mbps range it is recommended to set EgressRate = 0x0FA (to select 1 Mbps steps) and use EgressDec to select the number of 1 Mbps steps to use. For example: set EgressDec to 0x01 for 1 Mbps, set it to 0x02 for 2 Mbps, set it to 0x03 for 3 Mbps, etc.
- When egress rate shaping in the 100 Mbps to 1000 Mbps range it is recommended to set EgressRate = 0x019 (to select 10 Mbps steps) and use EgressDec to select the number of 10 Mbps steps to use. For example: set EgressDec to 0x0B for 110 Mbps, set it to 0x0C for 120 Mbps, set it to 0x0D for 130 Mbps, etc.

For Example: Set the egress rate to 256 k bits-per-second

EgressRate = 3906.25 or 0xF42 (see NOTEs above)

EgressDec = 0x04 (256 k / 64 k = 4)

Due to the cropping of fractions, the actual rate will be:

ActualRate = 8 bits * EgressDec / (32 ns * EgressRate)

ActualRate = 32 / (0.000000032 * 3906)

ActualRate = 256.016 bits-per-second

Note

If the desired rate is a ‘not to exceed’ rate then increment the EgressRate register value by 1 if there is a non-zero fraction in the decimal result. In the above example a value of 0xF43 for would be needed for a ‘not to exceed’ rate of 256 k bits-per-second. In this case, the ActualRate will be 255.950 kbps.

5.8.5.3 Frame Overhead Egress Shaping

When a port is configured in one of the Bits-per-Second egress rate shaping modes (Section 5.8.5.2) an optional overhead count, which follows the IEEE 802.3ar proposals, can be added to each frame. This fixed overhead is designed to adjust the egress rate taking into account frame modifications that may occur upstream from this device, such as adding a tag to the frame.

Frame Overhead (Port offset 0x09) can add a fixed delay to each frame in the amount of 0 to 60 bytes in 4 byte increments. The Frame Overhead byte delay is added to the bytes that were counted in the frame (Count Mode – Port offset 0x0A) to adjust the egress rate shaping accordingly.

Section 6. Provider Mode Ports

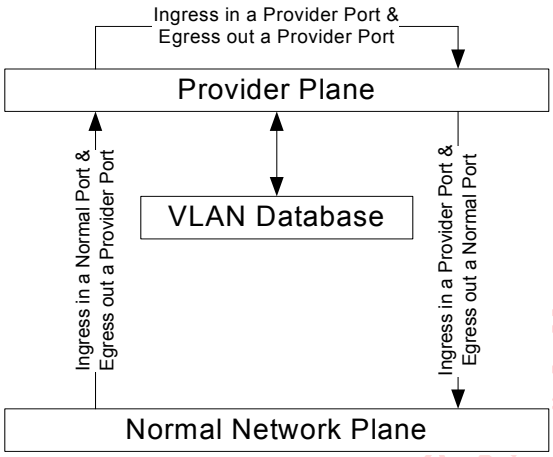
The discussions that follow assume that Provider ports are set in Provider mode (i.e., FrameMode = 0x2 at Port offset 0x04) and that Customer ports are set in Normal Network mode (i.e., FrameMode = 0x0 at Port offset 0x04). Provider mode is sometimes referred to as Double Tagging, Q-in-Q, or IEEE 802.1ad.

Provider ports are designed to bridge between Customer ports and Provider ports and visa versa. More than one Provider port can be supported (in a single device or between multiple devices interconnected with DSA ports – Section 7).

Notes

- When one or more Provider ports exist on the switch, Customer ports are expected to be Normal Network ports as defined in Section 5 although 802.1Q should not be enabled on Customer ports. Instead Customer ports should use the Port Based VLAN mode if port to port isolation is required for Customer to Customer data flow separation (Section 5.2.1). The Customer ports' EgressMode should also be set to Transmit Unmodified (Port offset 0x04 and Section 5.8.4). These restrictions occur because the 802.1Q VLAN Database is being used by the Provider Port(s) for S-TAG (Service Tag) switching and therefore it is not available for C-TAG (Customer Tag) switching and dynamic egress tag modifications. Per VLAN Spanning Tree (IEEE 802.1s – Section 5.2.3) is not available on the Customer ports for the same reason (the VTU and STU are being used by the Provider port(s)).
- Provider Ports must have their EgressMode set to 0x0 (transmit unmodified, Port offset 0x04).

Figure 38: Provider vs. Normal Data Planes

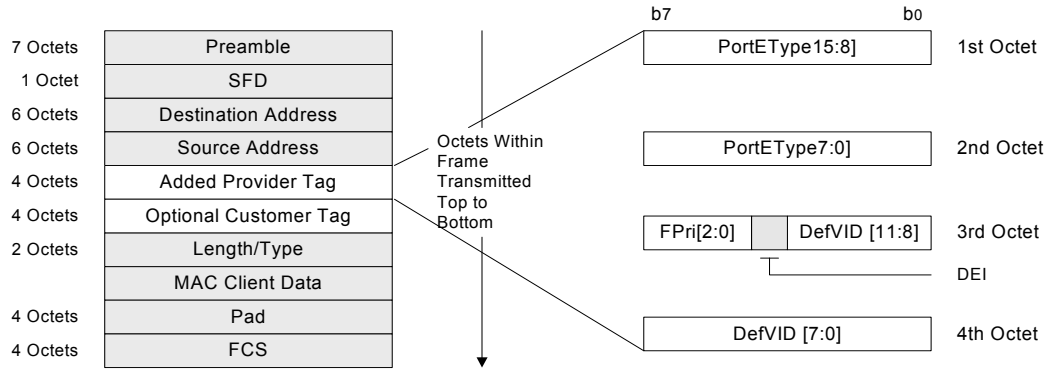


6.1 Customer to Provider

As stated in the NOTE above, Customer ports are expected to have their 802.1Q Mode Disabled (Port offset 0x08). This causes the ingress port's DefaultVID (Port offset 0x07) to be assigned to all ingressing frames as the Customer's S-VID regardless if the Customer's frame is tagged or untagged. The frame is then port based VLAN limited (by software configuration – Section 5.2.1) to egress only the correct ports, for example, only to the Provider Port(s). All the other Normal Network features of the device (as described in Section 4 and Section 5) are available to Customer ports, including Layer 2 PCLs (Section 5.1.3), QoS mapping (Section 5.4) and ingress rate limiting (Section 5.5), to name a few.

When the Customer's frame egresses a Provider Port the frame will egress with an S-TAG (service tag) added in front of the Customer's Tag, if one existed in the frame (Figure 39).

Figure 39: Provider Tag Format



The Ether type of the added S-TAG (1st and 2nd octet) comes from the egress port's PortEType register (Port offset 0x0F). The PRI bits (3rd octet) comes from the FPri assigned to the frame during ingress (Section 5.4) and the DEI (Drop Eligible Indicator) is set to zero. The S-VID (3rd and 4th octets) is set to the VID assigned to the frame during ingress, which will be the source port's DefaultVID (Port offset 0x07) if 802.1Q Mode is Disabled on the source port[1] (Port offset 0x08).

Note

The action described above occurs the same way regardless if the Customer port and Provider port are in the same physical switch device, or if they are in separate switch devices interconnected with DSA ports (Section 7).

6.2 Provider to Customer

When a frame enters a Provider Port, its Ether type (1st and 2nd octet in Figure 39) is compared to the Provider port's PortEType register (Port offset 0x0F). If a match exists, the frame is considered Provider Tagged (i.e., it has an S-TAG) and the frame's S-VID (3rd and 4th octets) is assigned as the frame's VID inside the switch (this cannot be overridden by the Provider port's ForceDefaultVID bit, Section 5.2.2.6, as a Provider Port is trusted – but read on below how the ForceDefaultVID bit may be used on Provider Ports). The ether type match also causes the frame's S-PRI (3rd octet) bits to be assigned as the frame's initial FPri. The 4-byte S-TAG is then removed from the frame (during ingress), getting it ready to be transmitted out a Customer port[2]. A new CRC is placed on the frame due to the removal of the S-TAG[3]. The frame is stored in the switch's memory looking identical to the way the Customer sent it into the switch, i.e., it is unmodified as far as the customer is concerned.

If the ingressing frame's ether type does not match the Provider port's PortEType register, the frame is not considered Provider Tagged. In this case the frame is processed as if it entered a Normal Network port getting a VID,

1. Or it will be the S-VID from the frame's S-TAG if the came in another Provider Port (see Section 6.4).
2. Frames are padded back up with zeros just before the CRC if the frame is less than 64 bytes in size due to the removal of an S-TAG.
3. Ingressing frames with a CRC error are discarded, therefore a bad frame cannot be made good by adding the new CRC.

FPri and QPri assigned to it as defined in Section 5, with one exception. The Provider port's DiscardTagged and DiscardUntagged bits (Port offset 0x08) work differently as follows:

- DiscardTagged will discard Provider Tagged frames with a non-zero S-VID
- DiscardUntagged will discard non-Provider Tagged frames, including Priority Only Provider Tagged frames whose S-VID = 0x000 and raw Customer frames, tagged or untagged

The DiscardUntagged function on Provider Ports can be used to ensure only 'good' Provider Tagged frame are allowed to enter the Provider Port. If however, non-Provider Tagged frames are allowed to enter the switch (by clearing the port's DiscardUntagged bit to zero) the Customer port(s) the frame is allowed to egress can be limited by setting the Provider Port's ForceDefaultVID bit to a one (Port offset 0x07) and setting the port's DefaultVID to an unused value. A DefaultVID of 0x000 will work here as the VID of 0x000 exists in the VTU just like all the other VID values. The ForceDefaultVID feature (Section 5.2.2.6) will only take effect on a Provider Port if the ingressing frame is not considered Provider Tagged (i.e., its ether type does not match the port's PortEType register). This DefaultVID assigned to the non-Provider frames will not show up in the frame as long as the MemberTag bits for this VID are set to egress unmodified (see Section 6.2.1).

6.2.1 Provider VID Processing

The VID assigned to the frame (the S-TAG's S-VID or the port's DefaultVID in case the frame was not properly Provider Tagged, see above) is used to access the VLAN Database assuming 802.1Q Mode (Port offset 0x08) is enabled on the Provider Port (Section 5.2.2). VLAN switching is needed to get each provider frame mapped to the correct Customer Port(s) while at the same time ensuring the frame does not get to any of the wrong Customers. The VTU's MemberTag bits for the Customer port(s) for each VID entry should be set to either:

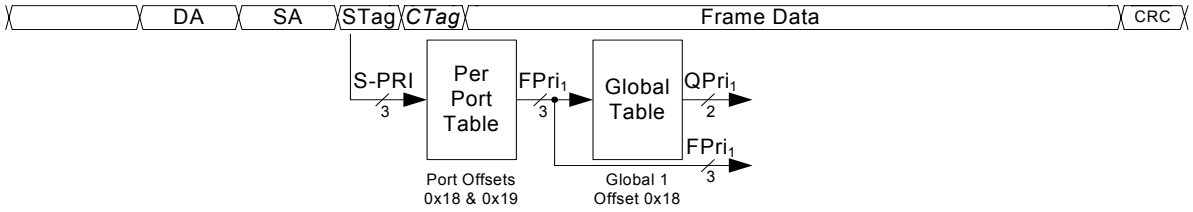
- Port is not a member of this VLAN, or
- Port is a member of this VLAN and frames are to egress unmodified

Since the frame's S-TAG was removed during ingress, and it already in the correct format for the Customer, using any other MemberTag mode will not work correctly.

6.2.2 Provider QoS Processing

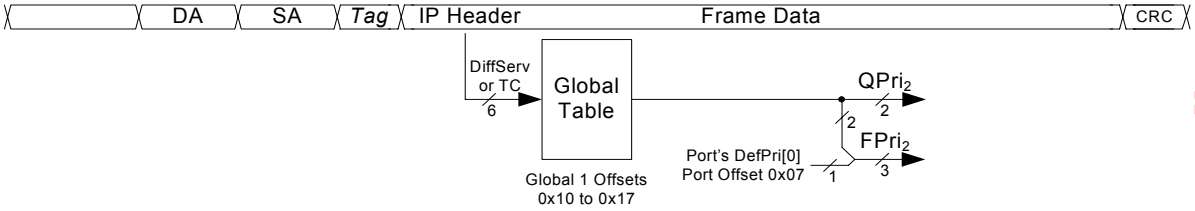
When the ingressing frame is considered Provider Tagged (i.e., its ether type matches the port's PortEType register) its S-PRI bits from the frame's removed STAG are used to define FPri1 and QPri1 as shown in Figure 40 below. This no different from the way Normal Network ports work (Section 5.4.1), i.e., the frame's 1st Tag is used. This means that any data from the frame's C-TAG (the Customer's Tag that follows the S-TAG, if one exists) cannot be used for QoS determination. It is assumed the S-Tag's priority can be trusted as it came from the Provider, and this is the proper priority to use inside this switch as this switch is still 'owned' by the Provider, giving service to many Customers. The only way to guarantee this service is to control it.

Figure 40: Provider S-PRI Remapping



Since the S-TAG is removed during ingress, the IP portion of the frame can still be used to determine FPri2 and QPri2 as shown in Figure 41. This is not recommended unless the IP priorities can be trusted.

Figure 41: Provider IP PRI Mapping



The rest of the QoS selection on Provider ports is the same as described in Section 5.4 (i.e., Initial Priority Selection, Frame Type Priority Overrides and Layer 2 Priority Overrides can be done).

If the ingressing frame is not considered Provider Tagged (i.e., the frame's Ether type does not match the Provider port's PortEType register) the QoS of the frame is determined as if it entered a Normal Network port getting FPri and QPri assigned to it as defined in Section 5.4.

6.3 Customer to Customer

Multiple Customer ports owned by the same Customer can be supported. If any single customer has more than one port[1] on the Provider switch, frame traffic between these Customer ports can occur directly, without the need of sending these frames to the Provider. This can be accomplished by expanding the Customer ports' Port Based VLANs (both in-chip and cross-chip – Section 5.2.1 and Section 7.5.3) to include all ports owned by the same Customer. By assigning all ports going to the same Customer with the same DefaultVID (Port offset 0x07) and the same FID (Section 4.4.8), the address database can be used to direct the frames to the correct port(s).

Any QoS priority overriding (including FPri changes) that was done inside the switch for frames going to the Provider port will not take effect outside the switch when the frame goes to another Customer port, assuming all Customer port's are configured to transmit frames unmodified (as defined by the Customer port's EgressMode bits, Port offset 0x04 for Customer to Customer flows, and by the MemberTag bits for the Customer's port as loaded into the VTU for the Customer's S-VID, Section 9.2.1, for Provider to Customer flows). This is the proper operation of a provider switch, i.e., to provide a 'pipe' for Customer data to flow through without the Customer data getting modified in any way.

Dynamic 802.1Q VLAN switching, i.e., limiting which ports the frame can go out, based upon the frame's Tag (C-Tag in this case) cannot be done for the Customer to Customer flows as the VLAN database is needed by the Pro-

1. This discussion is for non-Link Aggregated ports. Link Aggregated ports to a Customer are also supported using the standard rules for Port Trunks (Section 8.10).

vider Port(s) and its not available for the Customer ports (see Figure 38). As stated in Section 6.1, 802.1Q Mode should be disabled on Customer ports.

6.4 Provider to Provider

More than one Provider port going to the same or different Providers is supported. If any single Provider has more than one port[1] on the Provider switch, frame traffic between these Provider ports can occur directly, with dynamic 802.1Q VLAN switching if needed. This is accomplished automatically (both in-chip and cross-chip). During ingress, the Provider's S-TAG is removed from the frame (as defined in Section 6.2), with its S-VID being assigned as the frame's VID. If this frame is mapped to another Provider port, that port will insert a new Provider S-TAG as the frame egresses following the rules defined in Section 6.1. In this case the VID assigned during ingress will be the S-VID extracted from the frame when it ingressed the Provider Port.

The Ether type added to the egressing frame will come from the egress port's PortEType register (Port offset 0x0F). This means that each Provider port can support the same or different Ether types. Two ports on the device can be used as a full wire speed Ether type translator from one Provider's Ether type space to an alternate (Provider's) Ether type space. This occurs because the S-TAG from one Provider is always removed from the frame during ingress using that provider's Ether type and the alternate Provider's Ether type is added to the frame's S-TAG during egress.

Any QoS priority overriding (including FPri changes) that was done inside the switch for frames going to another Provider port will be updated in the frame when the frame goes out another Provider port as all Provider port's will add an S-TAG with an S-PRI using the FPri assigned to the frame during ingress.

Dynamic VLAN switching, i.e., limiting which ports the frame can go out, based upon the frame's Tag (S-Tag in this case) is done for the Provider to Provider flows (as its done on Provider to Customer flows) as 802.1Q should be enabled on the Provider Port(s).

The VID assigned to the frame (the S-TAG's S-VID or the port's DefaultVID in case the frame was not properly Provider Tagged, see Section 6.2) is used to access the VLAN Database assuming 802.1Q Mode (Port offset 0x08) is enabled on the Provider Port(s) (Section 5.2.2). VLAN switching is needed to get each provider frame mapped to the correct Provider and/or Customer Port(s) while at the same time ensuring the frame does not get to any of the wrong Customers and/or Providers. The VTU's MemberTag bits for any alternate Provider port(s) for each VID entry should be set to either:

- Port is not a member of this VLAN, or
- Port is a member of this VLAN and frames are to egress unmodified

Since the frame's S-TAG was removed during ingress, it is already in the correct format to get a new S-TAG added to it. This is done using the 'egress unmodified' MemberTag mode. Using any other MemberTag mode is reserved for future use and may not work correctly.

6.5 Recursive Provider Tag Stripping

Recursive Provider tag stripping is supported by default on all Provider ports, but it can be globally disabled by setting the Remove1PTag bit to a one (Global 2, offset 0x05).

Recursive Provider tag stripping removes one or more valid S-TAGs found in a frame as it ingresses a Provider port. A valid S-TAG is any S-TAG that has an Ether type that matches the port's PortEType register. The VID and

1. This discussion is for non-Link Aggregated ports. Link Aggregated ports to a Provider are also supported using the standard rules for Port Trunks (Section 8.10).

FPri assigned to the frame for switch processing (Section 6.2) always comes from the first, or outer, S-TAG that is removed. The content of all subsequent S-TAGs is ignored. Frames are padded back up with zeros just before the CRC if the frame is less than 64 bytes in size due to the removal of an S-TAG[1].

Recursive Provider tag stripping cannot occur, even if enabled, if the Provider port's PortEType register = 0x8100 as this is the same ether type used in Customer Tags and these must not be removed.

6.6 Restrictions on Provider Ports

All the Normal Network port functions described in Section 4 and Section 5 are available to Provider ports with the exception of any function that uses the port's PortEType register (as this register is needed to determine the Provider frame's Ether type on Provider Ports). Specific functions that should not be used on Provider ports:

- Don't use Layer 2 Policy Controls using the frame's Ether type (Section 5.1.3.2).
- Don't use Frame Type Priority Override using the frame's Ether type (Section 5.4.5).
- Don't use a Transmit Tagged or Transmit Untagging Egress Mode. Always use Transmit Unmodified[2] (Section 5.8.4).
- Don't used a Provider Port as a destination port for Mirrors[3] (Section 5.1.3 for Layer 2 Mirrors and Section 8.9 for Ingress and/or Egress Mirrors).

6.7 Restrictions on Customer Ports

All the Normal Network port functions described in Section 4 and Section 5 are available to Customer ports with the exception of any function that uses the VTU (as the VTU's contents are needed for the Provider ports, see Section 6.2). Specific functions that should not be used on Customer ports when one or more Provider ports exist on the switch are:

- Don't enable any of the 802.1Q VLAN modes (Section 5.2.2). Use Port Based VLANs only.
- Don't use 802.1s Per VLAN Spanning Tree (Section 5.2.3).
- Don't use Layer 2 Priority Override using the frame's Customer VID (Section 5.4.6).
- Don't use a Transmit Tagged or Transmit Untagging Egress Mode. Always use Transmit Unmodified[4] (Section 5.8.4).

1. Ingressing frames with a CRC error are discarded, therefore a bad frame cannot be made good by adding the new CRC.
2. Provider ports get an S-TAG added automatically based upon FrameMode (Port offset 0x04) instead of using any other indicator. Therefore, Transmit Unmodified will do what is needed and is the correct EgressMode to use.
3. Mirrors cause a frame to be replicated, the original frame and then a mirrored or copied control frame. If these copied frames egress a Provider port there is no way for the provider to distinguish which frame was the original frame and which one was the copy. But a management CPU can. An alternate approach is to use a dedicated port for the mirrored data that contains only the copies (see Section 8.9).
4. See Section 6.2.1 for the reasons why Transmit Unmodified is the correct EgressMode to use on Customer ports.

Section 7. Distributed Switch Architecture (DSA) Ports

The discussions that follow assume that ports used to interconnect devices to each other and to the management CPU (referred to as internal ports) are set in Distributed Switch Architecture (DSA) Tag mode (i.e., FrameMode = 0x1 at Port offset 0x04, except for Ether type DSA Tag, Section 7.9, where FrameMode = 0x3) while external ports are set to either Normal Network mode (Section 5) or Provider mode (Section 6).

An alternate DSA Tag mode, called Ether type DSA Tag (Section 7.9) is also supported. Ether type DSA encapsulates the standard DSA Tag after a programmable Ether type. The Ether type DSA mode is optimized for switch to CPU interconnections while the standard DSA is optimized as a chip-to-chip switch fabric extension.

Standard DSA Tag ports must have their EgressMode set to 0x0 (transmit unmodified, Port offset 0x04). Ether type DSA Tag ports can use any EgressMode setting (Section 7.9).

There are four major DSA Tag mode frame types:

- Forward – for normal frames
- To_CPU – for MGMT (management) or control frames that needs to go to the CPU
- From_CPU – for MGMT (management) or control frames that come from the CPU
- To_Sniffer – for MGMT (management) or control frames used for chip-to-chip communication

7.1 Forward DSA Tag

The Forward DSA Tag is applied to a frame as it egresses a DSA Tag port if the frame is not a special frame (e.g., it is not a MGMT frame – Section 8.3). It is the kind of DSA Tag that a CPU should use when sending a frame into the switch where the CPU wants the switch is to process the frame and figure out where it should go. Normal ingress and egress filtering rules apply to these frames, i.e., the frames are processes as if they entered a Normal Network port (Section 4 and Section 5). Most of the frames that go through a DSA Tag port contain the Forward DSA Tag format, defined in Figure 42 and Table 32.

Figure 42: Forward DSA Tag Format

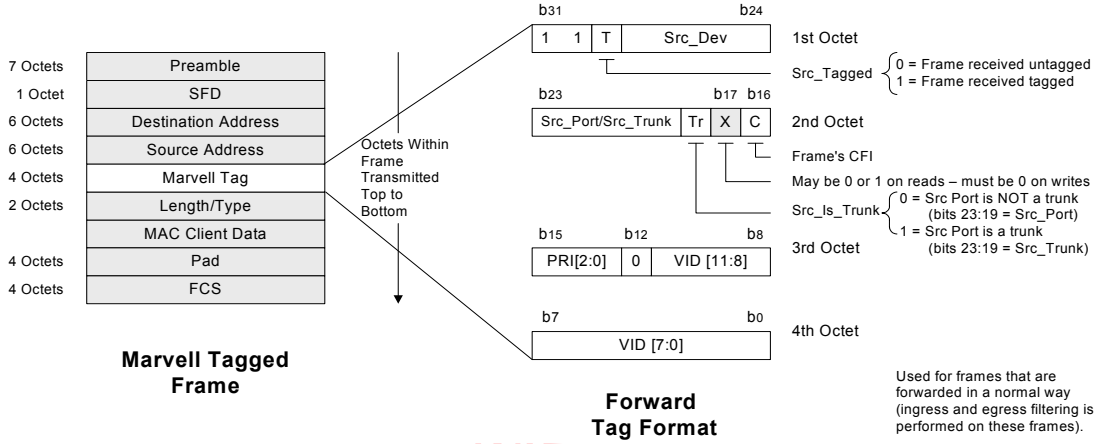


Table 32: Forward DSA Tag Fields

Frame's Field	Description
Src_Tagged	Source Tag Mode, i.e., the DSA Tag is placed in the frames on top of the standard IEEE Tag that was in the frame. Standard IEEE Tags contain an 0x8100 Ether type. This bit can only be set to a one if the frame came from a Normal Network port (Section 5) where 802.1Q is enabled (Section 5.2.2) and if the frame was IEEE Tagged. This bit will never be set to a one if the frame came in on a Provider Port (Section 6) nor if it entered a Normal Network port where 802.1Q Mode is Disabled even if the frame is IEEE Tagged (i.e., Port Based VLANs are being used). In these cases the DSA Tag is added to the frame leaving the rest of the frame's contents intact. This bit is used to tell the egress logic on Normal Network ports how to convert the frame from DSA Tag to Normal Network (see Table 33). The bit is ignored if the frame egresses a Provider Port as a DSA Tag is always converted into a Provider Tag with the Provider port's PortEType register being used as the Tag's Ether type[1].
Src_Dev	Source Device. These bits are used to define the original source device's number where the frame first ingressed (i.e., the first device where the frame Ingressed from a Normal Network (Section 5) or Provider port (Section 6) before being switched to an Internal DSA Tag port). These bits come from the source device's DeviceNumber register (Global 1 offset 0x1C).
Src_Port/Src_Trunk	Source Port or Source Trunk. If the Src_Is_Trunk bit, below, is zero, these bits are used to define the original source port's number (on the source device above). 0x00 indicates Port 0, 0x01 indicates Port 1, 0x02 for Port 2, 0x03 for Port 3 etc. If the Src_Is_Trunk bit, below, is one, these bits are used to define the Trunk ID of the 1st trunk this frame entered or passed through.
Src_Is_Trunk	Source is Trunk. When this bit is zero, it indicates this frame originally entered a non-trunked port and this frame has never passed through a trunked port. In this case, the Src_Port/Src_Trunk bits define the Src_Port. When this bit is one it indicates this frame originally entered a trunked port or this frame passed through a DSA port that was trunked. In this case, the Src_Port/Src_Trunk bits define the Src_Trunk.
C	The original frame's CFI (Canonical Format Indicator) bit if the frame was IEEE tagged when it originally entered a Normal Network port (Section 5) on the switch. It is the original frame's DEI (Drop Eligible Indicator) bit if the frame was Provider tagged when it originally entered a Provider port (Section 6) on the switch.
PRI[2:0]	The frame's FPri priority as determined by the Ingress rules of the last devices that this frame entered (Section 5.4 for Normal Network ports and Section 6.2.2 for Provider ports).
VID[11:0]	The frame's VLAN identifier as determined by the Ingress rules of the last devices that this frame entered (Section 5.2.2.5 for Normal Network ports and Section 6.2.1 for Provider ports).

1. In a properly configured Provider switch, this bit will never be a one because it will never be a one if the frame came from a Provider Port and is should never be a one if the frame came from a Customer Port since all Customer ports should have 802.1Q Mode set to Disable.

Table 33: Src_Tagged vs. Normal Network Egress Mode Actions

Frame's Src_Tagged	Port's Tagging Mode[1]	Comments
0	Unmodified	DSA Tag is removed from the frame.
1	Unmodified	DSA Tag is converted to IEEE Tag with an 0x8100 ether type.
0	Untagged	DSA Tag is removed from the frame.
1	Untagged	DSA Tag is removed from the frame.
0	Tagged	DSA Tag is converted to IEEE Tag with an 0x8100 ether type.
1	Tagged	DSA Tag is converted to IEEE Tag with an 0x8100 ether type.

1. As defined in Section 5.8.4.

Note

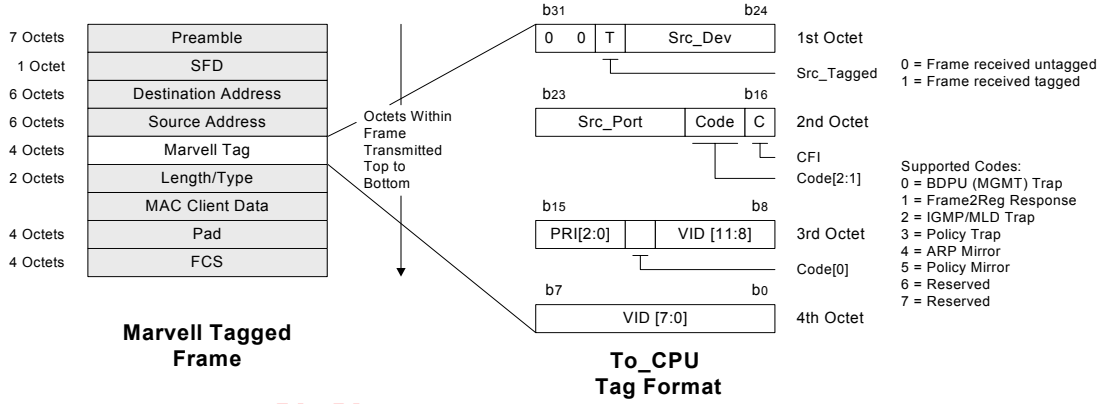
When the CPU sends a frame into the local switch device using the Forward DSA Tag format, it should set the Src_Dev to the value of the DeviceNumber (Global 1 offset 0x1C) of the device it is attached to and it should set the Src_Port equal to the port number on the device it is attached to. Bit 17 and bit 12 in Figure 42 must be cleared to zero. This way the Src_Dev and Src_Port values will match the physical port where the frames entered the switch. This will make all the features that used these fields in the frame more consistent (like cross-chip Port Based VLANs – Section 7.5.3).

7.2 To_CPU DSA Tag

When the CPU is running the Spanning Tree Protocol (Section 8.2) or it needs the switch to perform IGMP/MLD Snooping (Section 5.3.4), ARP Mirroring (Section 5.3.3) or Layer 2 Policy (Section 5.1.3), the CPU must receive some special frames. When the switch device is configured to detect special frames and they are also configured to egress these frames from a DSA Tag port (like to the CPU's port), the frame is modified with a To_CPU DSA Tag as it egresses the port. The format of the To_CPU Tag is defined in Figure 43 and Table 34.

In an environment where more than one device is connected together using DSA Tag ports, the device(s) in the middle will receive To_CPU DSA Tag frames. These frames are sent through the intermediate switch unmodified. They egress from the port defined by the CPUDest register (Global 1 offset 0x1A) unless the frame's CPU Code = 0x5 where it will egress from the port defined by the MirrorDest register instead (also in Global 1 offset 0x1A). This allows Layer 2 Policy Mirrors (Section 5.1.3) to be directed to some other port from the one the management CPU is connected to. In this way, a Normal Network port can be the final destination for Layer 2 Policy Mirrors.

Figure 43: To_CPU DSA Tag Format



Notes

- If a CPU sends a To_CPU DSA Tag frame into the switch, it will be mapped as stated above, which means the CPU will likely receive the frame back unmodified (depending upon the frame's CPU Code and the setting of the CPUDest and MirrorDest registers). This can be used as a diagnostic to test the flow of frames between the CPU and the switch. If the CPU needs to send frames out a specific port, use the From_CPU DSA Tag frame format instead.
- To_CPU DSA Tag frames are considered MGMT (management) frames. MGMT frames are processed differently inside the switch. See Section 7.6.

Table 34: To_CPU DSA Tag Fields

Frame's Field	Description
Src_Tagged	Source Tag Mode, i.e., the DSA Tag is placed in the frames on top of the standard IEEE Tag that was in the frame. Standard IEEE Tags contain an 0x8100 Ether type. This bit can only be set to a one if the frame came from a Normal Network port (Section 5) where 802.1Q is enabled (Section 5.2.2) and if the frame was IEEE Tagged. This bit will never be set to a one if the frame came in on a Provider Port (Section 6) nor if it entered a Normal Network port where 802.1Q Mode is Disabled even if the frame is IEEE Tagged (i.e., Port Based VLANs are being used). In these cases the DSA Tag is added to the frame leaving the rest of the frame's contents intact.
Src_Dev	Source Device. These bits are used to define the original source device's number where the frame first ingressed (i.e., the first device where the frame Ingressed from a Normal Network (Section 5) or Provider port (Section 6) before being switched to an Internal DSA Tag port). These bits come from the source device's DeviceNumber register (Global 1 offset 0x1C).
Src_Port	Source Port. These bits are used to define the original source port's number (on the source device above). 0x00 indicates Port 0, 0x01 indicates Port 1, 0x02 for Port 2, 0x03 for Port 3 etc. These bits always reflect the physical Src_Port of To_CPU frames even if the physical Src_Port is a Trunk port (Section 8.10).
Code	To_CPU frame type code. These bits are set by the original Src_Dev (see above) to indicate the kind of To_CPU frame. The device generates a frame type code depending upon the reason as defined in Table 35.
C	The original frame's CFI (Canonical Format Indicator) bit if the frame was IEEE tagged when it originally entered a Normal Network port (Section 5) on the switch. It is the original frame's DEI (Drop Eligible Indicator) bit if the frame was Provider tagged when it originally entered a Provider port (Section 6) on the switch.
PRI[2:0]	The frame's FPri priority as determined by the Ingress rules of the last devices that this frame entered (Section 5.4 for Normal Network ports and Section 6.2.2 for Provider ports).
VID[11:0]	The frame's VLAN identifier as determined by the Ingress rules of the last devices that this frame entered (Section 5.2.2.5 for Normal Network ports and Section 6.2.1 for Provider ports).

Table 35: To_CPU Code Meanings

Code	Name	Comments
0x0	MGMT Trap	Placed on re-directed (trapped) frames that come from DA MGMT Trapping (Section 8.3)
0x1	Frame2Reg	Placed on Remote Management response frames (Section 10)
0x2	IGMP/MLD Trap	Placed on re-directed (trapped) frames that come from IGMP/MLD Trapping (Section 5.3.4)
0x3	Policy Trap	Placed on re-directed (trapped frames that come from Layer 2 Policy Traps (88E6097 only - Section 5.1.3)
0x4	ARP Mirror	Placed on mirrored or copied frames that come from ARP Mirroring (Section 5.3.3)
0x5	Policy Mirror	Placed on mirrored or copied frames that come from Layer 2 Policy Mirrors (88E6097 only - Section 5.1.3)
0x6	Reserved	Reserved for future use.
0x7	Reserved	Reserved for future use.

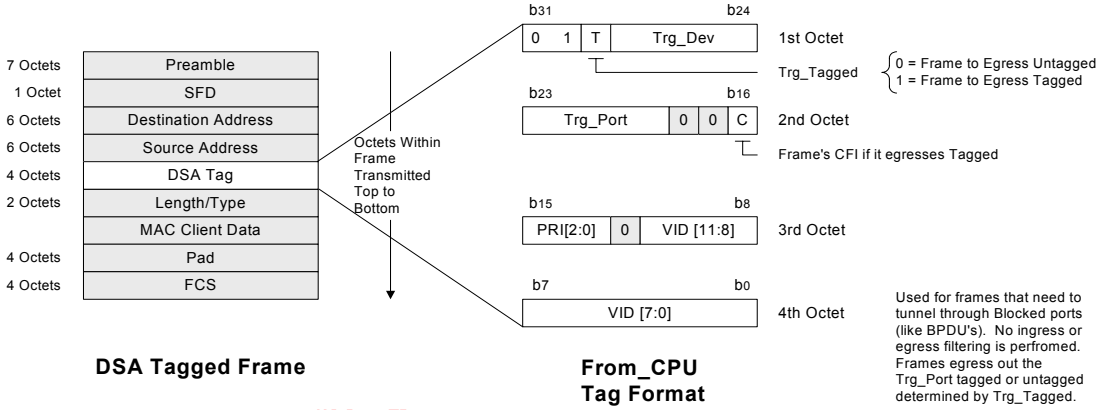
7.3 From_CPU DSA Tags

When the CPU is running the Spanning Tree Protocol (Section 8.2), or other protocols, it needs to be able to transmit special frames, like BPDU's, out any of the ports on the switch. The CPU can do this by using the From_CPU DSA Tag. The format of the From_CPU Tag is defined in Figure 44 and Table 36.

In an environment where more than one device is connected together using DSA Tag ports, the device(s) in the middle will receive From_CPU DSA Tag frames. In this case, the receiving port examines the frame's Target Device (Trg_Dev) field to see if this device is the target (by comparing the frame's Trg_Dev value to the local device's DeviceNumber register at Global 1 offset 0x1C). If this is the target device, the frame is sent out the port indicated in the frame's Target Port (Trg_Port) field. In this case, it is expected that the frame will be mapped to a Normal Network (Section 5) or a Provider port (Section 6) where the frame will egress the port IEEE Tagged or Untagged based on the frame's Trg_Tagged bit (see Table 36).

If this is not the target device, the frame is sent out the port programmed into the Device Mapping table (Global 2 offset 0x6) using the Trg_Dev as an index into the table. In this case it is expected that the frame will be mapped to another DSA Tag port where the frame will egress the port unmodified.

Figure 44: From_CPU DSA Tag Format



Notes

- From_CPU DSA Tag frames are considered MGMT (management) frames. MGMT frames are processed differently inside the switch – including the ability to egress Blocked ports (Table 26). See Section 7.6.
- The CFI bit in From_CPU DSA Tag frames is placed at b16. When this frame egresses out a normal network port the CFI bit will be moved to its correct position at b12 (which is the DEI bit on Provider Ports – Section 6).

Table 36: From_CPU DSA Tag Fields

Frame's Field	Description
Trg_Tagged	Target Tag Mode. This bit allows the CPU to define if this frame is to egress the target port IEEE Tagged or not. If this bit is set to a one, the frame egresses the target port with a proper IEEE 802.3ac Tag (i.e., the 1st two octets of the From_CPU DSA Tag are converted to the 0x8100 Ether type and the frame's C bit is moved between the PRI and VID bits). If this bit is cleared to a zero, the frame egresses the target port Untagged (i.e., the 4 byte DSA Tag will be removed from the frame). If the target port is a Provider Port (Section 6) see the NOTE below.
Trg_Dev	Target Device. These bits are used to define the target device's number. Use 0x00 for single chip switches (assuming the chip's DeviceNumber in Global 1 offset 0x1C = 0x00). From_CPU frames pass from chip to chip (using the switch port defined by the Device Mapping Table, Global 2 offset 0x06) until the frame finds a chip where the frame's Trg_Dev field matches the chip's DeviceNumber register.
Trg_Port	Target Port. These bits are used to define the target port's number (on the target device – see Trg_Dev above). Use 0x00 for Port 0, 0x01 for Port 1, 0x02 for Port 2, etc. From_CPU frames will Egress the Target Port on the Target Device (see above).
C	CFI bit. This bit is used as the frame's IEEE tag CFI bit if the frame egresses the target port Tagged (as defined by the Trg_Tagged bit above). If the target port is a Provider Port (Section 6) this bit will be the Provider Tag's DEI bit (see NOTE below).
PRI[2:0]	The frame's priority. PRI[2:1] are used to indicate which egress queue these frames are to be switched to unless overridden by the Priority Override Table (Global 2 offset 0x0F – Section 7.8). All three PRI bits are used as the frame's IEEE tag priority if the frame egresses the target port Tagged (as defined by the Trg_Tagged bit above).
VID[11:0]	The frame's VLAN identifier. These VID bits are ignored inside the switch on From_CPU frames. They are only used as the frame's IEEE VID if the frame egresses the target port Tagged (as defined by Trg_Tagged above).

Note

If the target port for a From_CPU frame is a Provider Port and if that Provider port's PortEType register (Port offset 0x0F) is not 0x8100 (i.e., the Provider Port is using a non-0x8100 Ether type) then have the CPU build the From_CPU DSA Tag with Trg_Tagged = 0 and place the desired Provider Tag (Ether type, PRI, DEI and VID) right after the From_CPU DSA Tag. When this frame egresses, the Provider Port the DSA Tag will be removed and the resulting frame will have the desired Provider Tag where it should be.

7.4 To_Sniffer DSA Tag

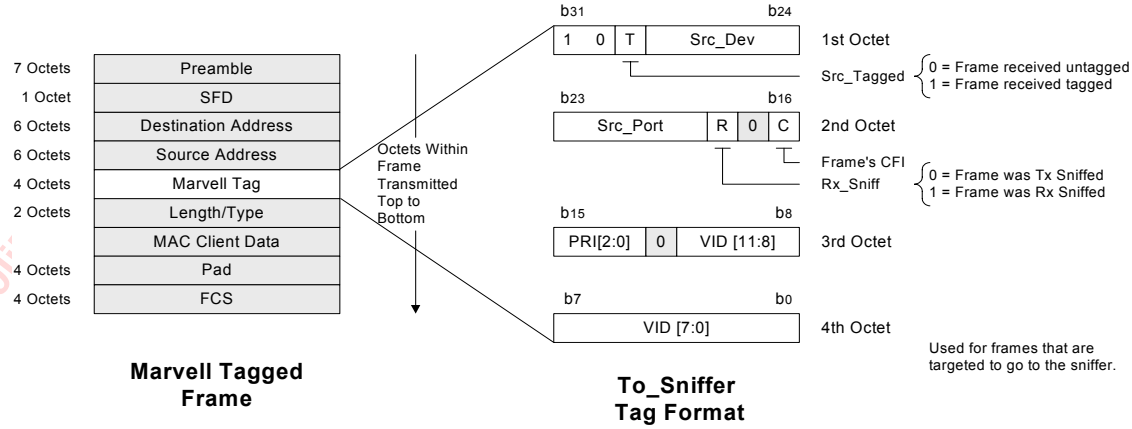
To_Sniffer DSA Tag frames are used to support chip-to-chip mirroring (Section 8.9). They are used to map Ingress Monitor Source (IMS) or Egress Monitor Source (EMS) frames to this device's Ingress Monitor Destination (IMD) or Egress Monitor Destination (EMD) port. Normally these frames do what they need to do without CPU processing. But if the CPU's port is the final IMD or EMD then the CPU will get these frames with the formats defined in Figure 45 and Table 37.

In an environment where more than one device is connected together using DSA Tag ports, the device(s) in the middle will receive To_Sniffer DSA Tag frames. These frames are sent through the intermediate switch unmodified. They egress from the port defined by the device's EgressMonitorDest register (Global 1 offset 0x1A) if the frame was Tx Sniffed (the R bit in Figure 45 = 0) or from the port defined by the device's IngressMonitorDest register (Global 1 offset 0x1A) if the frame was Rx Sniffed (the R bit in Figure 45 = 1).

Note

Provider Ports (Section 6) should not be the destination port for To_Sniffer frames, or any Mirror for that matter, since there is no way for the provider to distinguish these frames from the original frame data. On the other hand, a management CPU could be the destination for these frames because the DSA Tags will differentiate them from the normal frames.

Figure 45: To_Sniffer DSA Tag Format



Note

To_Sniffer DSA Tag frames are considered MGMT (management) frames. MGMT frames are processed differently inside the switch. See Section 7.6.

Table 37: From_CPU DSA Tag Fields

Frame's Field	Description
Src_Tagged	Source Tag Mode, i.e., the DSA Tag is placed in the frames on top of the standard IEEE Tag that was in the frame. Standard IEEE Tags contain an 0x8100 Ether type. This bit can only be set to a one if the frame came from a Normal Network port (Section 5) where 802.1Q is enabled (Section 5.2.2) and if the frame was IEEE Tagged. This bit will never be set to a one if the frame came in on a Provider Port (Section 6) nor if it entered a Normal Network port where 802.1Q Mode is Disabled even if the frame is IEEE Tagged (i.e., Port Based VLANs are being used). In these cases the DSA Tag is added to the frame leaving the rest of the frame's contents intact.
Src_Dev	Source Device. These bits are used to define the original source device's number where the frame first ingressed (i.e., the first device where the frame Ingressed from a Normal Network (Section 5) or Provider port (Section 6) before being switched to an Internal DSA Tag port). These bits come from the source device's DeviceNumber register (Global 1 offset 0x1C).
Src_Port	Source Port. These bits are used to define the original source port's number (on the source device above). 0x00 indicates Port 0, 0x01 indicates Port 1, 0x02 for Port 2, 0x03 for Port 3 etc. These bits always reflect the physical Src_Port of To_Sniffer frames even if the physical Src_Port is a Trunk port (Section 8.10).
Rx_Sniff	Receiver Sniff. This bit is set to a one if the frame came from an Ingress Monitor Source (IMS) port. This bit will be cleared to a zero if the frame came from an Egress Monitor Source (EMS) port. See Section 8.9.
C	The original frame's CFI (Canonical Format Indicator) bit if the frame was IEEE tagged when it originally entered a Normal Network port (Section 5) on the switch. It is the original frame's DEI (Drop Eligible Indicator) bit if the frame was Provider tagged when it originally entered a Provider port (Section 6) on the switch.
PRI[2:0]	The frame's FPri priority as determined by the Ingress rules of the last devices that this frame entered (Section 5.4 for Normal Network ports and Section 6.2.2 for Provider ports).
VID[11:0]	The frame's VLAN identifier as determined by the Ingress rules of the last devices that this frame entered (Section 5.2.2.5 for Normal Network ports and Section 6.2.1 for Provider ports).

7.5 Cross-chip Features Using DSA Links

The switch fabric of one device is extended when two or more devices are linked together with Distributed Switch Architecture (DSA) ports. All the features of the device are supported cross-chip like Trunking (Section 8.10) and Mirroring (Section 8.9) along with Flow Control and VLANs (described below).

7.5.1 Cross-chip Flow Control

In this document, flow control refers to a generic method used to prevent frame loss at the expense of latency and QoS. When flow control is enabled on a port, “back pressure” is the specific mechanism used on half-duplex ports, while full-duplex ports use the IEEE “PAUSE” based mechanism. When flow control is enabled, the queue controller will use a different algorithm such that frames are not dropped when congestion occurs. When flow control is disabled, the queue controller will use a tail drop mechanism such that lower priority frames get dropped when congestion occurs (Section 5.6).

Flow control is supported cross-chip, without head of line blocking. The mixing of port flow control modes is also supported. Some ports in the switch can be flow controlled based (i.e., flow control is enabled – Section 4.3.2, to Section 4.3.4), while others can be QoS based (i.e., flow control is disabled). When flow control enabled ports switch frames to other flow control enabled ports there will be no frame loss. This can occur at the same time that QoS enabled ports are switching frames to other QoS enabled ports. QoS actions will occur on these flows.

Cross-chip flow control must be enabled in a multi chip switch whenever more than one port is configured with flow control enabled. The following steps need to be done in each device:

1. Enable cross-chip flow control messages, by setting FlowControlMessage to a one (Global 2 offset 0x05).
2. Ensure ForceFlowControlPri is set to a one and FC Pri is set to 0x7 (Global 2 offset 0x05).
3. Ensure the Priority Override Table (Global 2 offset 0x0F – Section 7.8) has entry 0x9 either disabled or enabled with a QPri setting of 0x3.
4. Enable flow control on both sides of each DSA link that (but don’t enable flow control on the CPU’s port). On DSA links Flow control must be forced on (Section 4.3.4).
5. Enable LimitOut with a value of 0xFF (Port offset 0x02 – Section 4.3.5) on at least one side of each DSA link.
6. Use the default values in the Flow Control Delays register (Global 2 offset 0x04).
7. Each device must have a unique DeviceNumber (Global 1 offset 0x01C) and the Device Mapping Table (Global 2 offset 0x06) must be properly configured in each chip indicating what port to egress frames targeted to a particular DeviceNumber.

Note

If all ports have flow control disabled, it is best to disable flow control on the DSA links as well. If only one port is requested by the user to have flow control enabled, it is best not to enable flow control at all on any of the ports (flow control only works when both the ingress and the egress port have flow control enabled).

7.5.2 Cross-chip 802.1Q VLANs

802.1Q VLANs are supported cross-chip. Each device supports the full 4,096 VID’s in the VTU (Section 9.2), but each device only has storage for the port membership information for its own local ports. So when a VID is added with a defined port membership in one device, the same VID should be added to all devices in the switch defining the port membership for the local ports on each device (which could be the DSA port only). The DSA ports should be made members of all VIDs as they are extensions of the switch fabric and 802.1Q Secure and Check policy should not occur here (see Section 7.7). Either disable egress VID checking (Section 5.8.3) or ensure the Default-VID for all ports (Port offset 0x07) is contained in the VTU. 802.1Q needs to be enabled on the DSA ports as well so that the VID lookups for port membership will occur.

7.5.3 Cross-chip Port Based VLANs

The device supports a very flexible cross-chip port based VLAN system that is used for all non-MGMT frames even if 802.1Q is enabled on the port. The 512 x 11 entry cross-chip Port VLAN Table (shown in Figure 46) is used for this feature. The table is accessed using two registers at Global 2 offsets 0x0B and 0x0C.

Cross-chip port based VLANs are supported on Forward frames received on Distributed Switch Architecture (DSA) or Ether type DSA[1] ports only (see Frame Mode in Port offset 0x04 and Section 7.1). It is not applied to any other frame type that enters the port.

Forward frames contain source port information about the frame in its Src_Dev (Source Device) and Src_Port/ Src_Trunk (Source Port or Source Trunk) fields (Section 7.1). The Src_Is_Trunk field indicates the kind of data that is contained in the Src_Port/Src_Trunk field. This information in the DSA Forward frames indicates the Normal Network port or Provider port the frame originally entered when ingressing the collection of cascaded or stacked switch devices[2].

When a DSA Forward frame enters a DSA port[3] the frame's source port information is examined and used to access the cross-chip Port VLAN Table (Global 2 offsets 0x0B and 0x0C). The data found in the table indicates which port or ports, in this device, the frame can egress. A one in a port's bit position indicates the frame is allowed to egress that port. Port 0's bit is in bit 0, Port 1's bit is in bit 1, etc. DSA ports that connect to other switch devices should always have their port's bit set to a one in the table for all table entries as the purpose of the table is to limit which Normal Network (or Provider) ports the frame can egress based upon what Normal Network (or Provider) port it originally entered the switch on.

Two modes of accessing the Cross-chip Port VLAN Table are supported:

- When 5 Bit Port is cleared to a zero (Global 2, offset 0x1D) the table is configured to be used with Marvell® Link Street® devices. Since all current Marvell Link Street® devices support no more than 11 ports per device and no more than 16 trunks per system, only the lower 4 bits of the Src_Port/Scr_Trunk field is needed. In this mode the full 5 bits of the Src_Dev field is used. So the 9 bit Table Pointer is constructed as Src_Dev[4:0], Src_Port[3:0] (1st column of Figure 46).
- When 5 Bit Port is set to a one the table is configured to be used with Marvell DX or other devices where more than 16 ports or more than 16 trunks are used in the system. In this case the full 5 bits of the Src_Port/ Src_Trunk field is needed. In this mode, the Src_Dev field is reduced to 4 bits. So the 9 bit Table Pointer is constructed as Src_Dev[3:0], Src_Port[4:0] (2nd column of Figure 46).

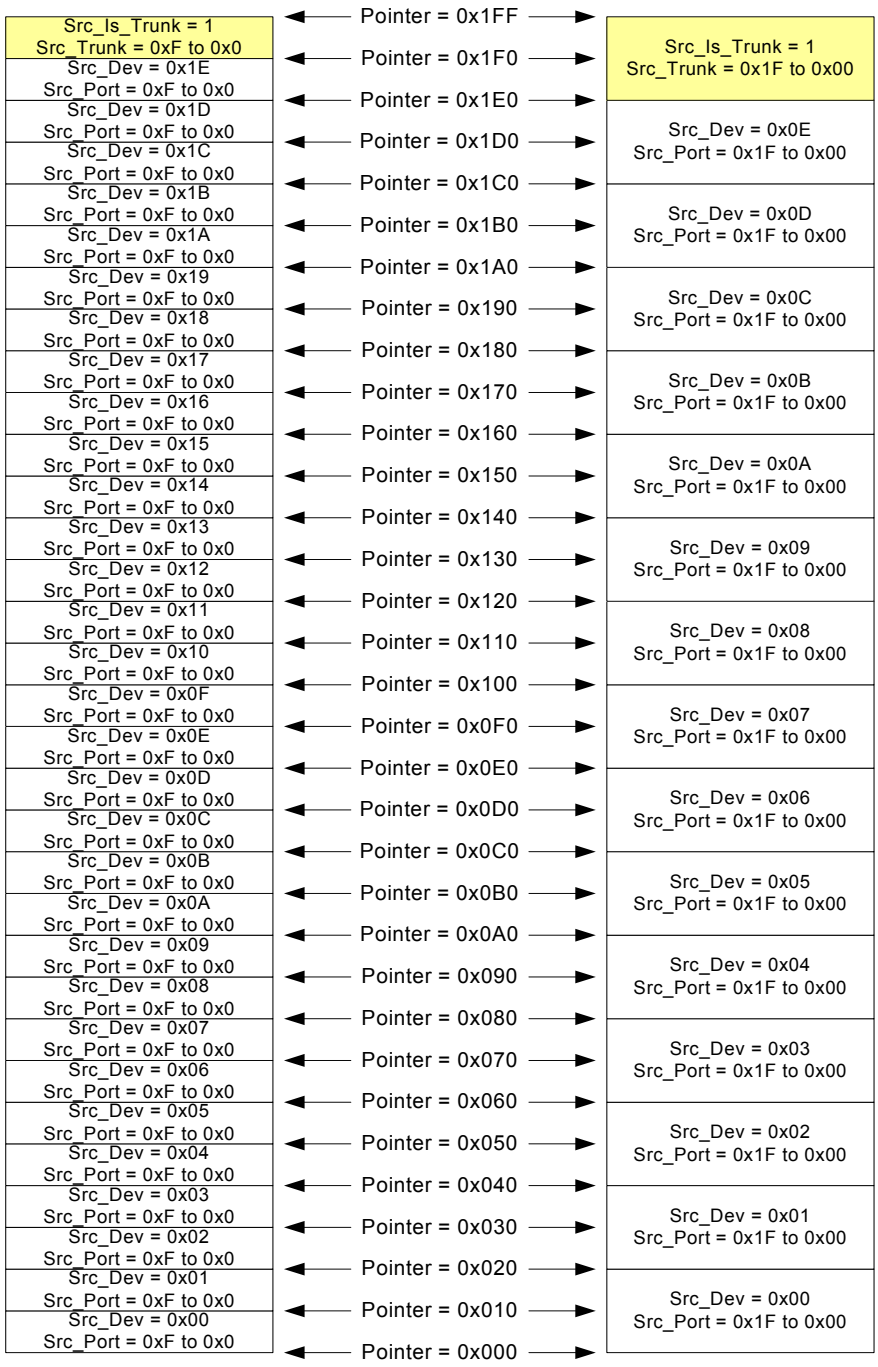
Cross-chip Trunk ports are supported by using the top 16 or 32 entries of the Cross-chip Port VLAN Table (the yellow boxes in Figure 46 when Src_Dev = 0x1F or 0x0F depending upon the value of 5 Bit Port – Global 2 offset 0x1D). This occurs when a DSA Forward frame enters a DSA port and the frame's Src_Is_Trunk = 1. Therefore, when this feature is used, Device Number 0x1F cannot be used when 5 Bit Port = 0, and Device Numbers 0x0F to 0x1F cannot be used when 5 Bit Port = 1.

The Cross-chip Port VLAN Table is used in conjunction with the In-Chip Port VLAN Map (Section 5.2.1.1) and 802.1Q VLANs if enabled (Section 5.2.2). If any of these VLAN functions masks (or prevents) a frame from egressing a port that frame will not be allowed to go out that port.

MUX'ing frames to a Router like a Marvell DX device (Section 8.8.1) is supported with the Cross-chip Port VLAN Table. The In-Chip Port VLAN Map on the frame's original source port (Section 5.2.1.1) MUX'es the frame to the Router's port. If the Router 'returns' the frame back to this original source device[4], the Cross-chip Port VLAN Table is then used to determine which ports the frame can egress.

1. If the port's Frame Mode is Ether type DSA, then the frame must be a Forward DSA Tag frame instead of a Normal Network frame or this feature will not be applied to the frame as Normal Network frames do not contain any physical source port information.
2. It is assumed that the collection of cascaded and/or stacked switch devices are interconnected using DSA Frame Mode links (Port offset 0x04).
3. Or Ether type DSA port
4. The connection between the device and the Router must be in DSA Frame Mode (Port offset 0x04).

Figure 46: Cross-chip Port VLAN Table Formats



7.6 Switch Handling of DSA MGMT Frames

As stated in Section 7.1, Forward DSA Tag frames are processed as if they entered a Normal Network port (Section 4 and Section 5). But the other three major DSA Tag types, To_CPU (Section 7.2), From_CPU (Section 7.3) and To_Sniffer (Section 7.4), are all considered MGMT (management) frames and they are processed differently as follows:

- The various frame types get mapped to their destination ports as defined in there respective sections. No other frame mapping functions occur. DA mapping, VID mapping, Port Based VLAN mapping, Trunk Load Balancing, Layer 2 Policy, etc. are all ignored. It does not matter if the ingress or egress port is Blocked (Section 5.1.1), these frames will go through.
- QoS priority overrides are ignored except for the Priority Override Table (Global 2 offset 0x0F) which is used for Secure Control Technology (Section 7.8).
- SA Filtering is ignored except for Drop On Unlock, i.e, frames from a potential hacker - see Note (Section 5.1.2.3).
- 802.1Q Secure and Check frame drop conditions are ignored (Section 5.2.2.8).
- Discard Untagged and Discard Tagged frame drop conditions are ignored (Section 5.2.2.3 and Section 5.2.2.4).
- Source addresses are not learned or refreshed (Section 4.3.3).

Note

DropOnUnlock (Section 5.1.2.3) will discard MGMT frames based on SA. This feature is not intended for DSA Ports. Instead it is intended for Normal Network or Provider Ports where frames can become MGMT as soon as they enter the switch port based on the frame's DA. Supporting DropOnUnlock in this case can help prevent BPDU (or any MGMT) DoS Attacks by discarding all MGMT frames from a particular source (or sources).

7.7 Proper Usage of DSA Tag Ports

Distributed Switch Architecture (DSA) Tag ports are used to interconnect devices to each other and to the management CPU. These internal ports are an extension of the switch fabric and therefore they are inherently trusted. DSA Tag ports typically carry frames from many different source ports (i.e., frames from potentially non-trusted external ports set to either Normal Network mode - Section 5, or Provider mode - Section 6).

The difference in the trust levels between the internal and external port types dictates the following usages (assuming multi-switch chip implementations):

- Policy (Section 5) needs to be done on the external ports (Normal Network or Provider) as the frames first enter the switch. This includes Denial of Service (Dos) attack prevention using PIRL resources (Section 5.5) on all frames that go to the CPU (MGMT, ARPs, IGMP, etc.).
- DSA Tag ports need to be an open pipe of data into the switch device. Therefore Policy must not be done on DSA Tag ports as that policy will affect multiple source and destination ports. This means that none of the feature described in Section 5 should be enabled on DSA Tag ports with the exception of VLAN mapping (Section 5.2.2.8).
- Basic switch operations (Section 4) with the exception of 802.1X (Section 4.4.7) must be done on DSA Tag ports so that learning and switching can occur in the local device. 802.1X should be done on the external ports only.

- Learn2All should be enabled for Cross-chip learning (Global 1 offset 0x0A). The MessagePort bit (Port offset 0x05) must be set on all DSA links (except for the CPUs link) for this to work.
- VLAN switch operations (the mapping portions of Section 5.2, not the Policy, i.e., frame dropping portions) must be done on DSA Tag ports so the local VLAN isolation can be done.
  - Use In-chip Port Based VLANs to prevent loops when SW_24P is enabled (Section 5.2.1.1).
  - Use Cross-chip Port Base VLANs (Section 7.5.3) when at least one external port is in a Port Based VLAN mode. All switch devices should have their Port VLAN Table filled for every external (source) port configured in this mode.
  - Use 802.1Q VLAN MemberTag mapping and tagging (Section 5.2.2) when at least one external port is using any of the three 802.1Q enabled modes. All switch devices should have every VID being used in the switch defined and loaded into their local VTU.
- Enable Cross-chip flow control if needed (Section 7.5.1).
- Force the link up on all SERDES or (G)MII DSA ports, including the CPU's port (Section 4.3.7 – Link will not come up automatically on internal ports).
- Mirroring (all types) can only be done on DSA Ports that are connected directly to a CPU. Chip-to-chip interconnected DSA ports must not enable any mirroring.
- Frame type priority overrides can be done for Secure Control Technology only (Section 7.8).
- Do not use the Ether type DSA frame (Section 7.9) format on chip-to-chip interconnections. It is designed to be used on the port directly connected to a CPU.
- Properly configure each device's Device Mapping Table (Global 2 offset 0x06 – Section 7.3).

7.8 Secure Control Technology (SCT)

Secure Control Technology (SCT) is designed to get Management (MGMT) frames to the CPU in a programmable priority order. In multi-switch chip systems there are two parts to this:

- Get the MGMT frames from a chip which is not directly connected to the CPU to the chip that is directly connected to the CPU. This is generally done by reserving and using QPri 3, the highest queue priority in the device, for this MGMT traffic. This approach does not work for mirrors, however.
- Distribute the MGMT frames into multiple egress queue priorities on the device directly connected to the CPU. Assuming the management CPU is isolated from all switch traffic that is not intended for the CPU (i.e., its not a member of any VLAN) then all four egress queue priorities are available on the port directly connected to the CPU.

The Priority Override Table (Global 2 offset 0x0F) is used for SCT. In switch devices that are not connected directly to the CPU, the table is set to QPri 3 for all MGMT except mirrors. In the switch device connected directly to the CPU, the table is set to the desired priority depending upon the MGMT type. Table 38 shows an example.

Table 38: Secure Control Technology Example

Frame Type	QPri for chips not connected to the CPU	QPri for chip connected to the CPU	Description
Multicast MGMT	0x3	0x2	Used on 802.1 protocols like Spanning Tree, etc.
Unicast MGMT	0x3	0x3	Can be used to get unicast frames to the CPU crossing VLANs and blocked ports.
Code = 0x1	0x3	0x3	Used on To_CPU Frame to Register response frames to the CPU.
Code = 0x2	0x3	0x2	Used on To_CPU IGMP/MLD Snoop traps to the CPU.
Code = 0x3	0x3	0x3	Used on To_CPU Layer 2 Policy traps to the CPU.
Code = 0x4	0x1	0x1	Used on ARP mirrors to the CPU (and elsewhere).
Code = 0x5	Don't Override	0x1	Used on Layer 2 Policy mirrors to the CPU (and elsewhere).
From_CPU	0x3	0x3	Used on frames sent into the switch by the CPU.
Flow Control	0x3	0x3	Used on cross-chip flow control messages (won't go to CPU Port)
To_Sniffer Tx	0x0	0x0	Used on Egress Monitor Source frames.
To_Sniffer Rx	0x0	0x0	Used on Ingress Monitor Source frames.
EType	0x2	0x2	Used on Ether type priority overrides (not used on DSA Ports)
Broadcast	0x0	0x0	Used on Broadcast priority overrides (not used on DSA Ports)

7.9 Ether Type DSA Tag

An alternate DSA Tag mode, called Ether type DSA Tag is supported and defined in Figure 47 and Table 39. Ether type DSA encapsulates the standard DSA Tags, described above, after a programmable Ether type. The Ether type DSA mode is optimized for switch to CPU interconnections for the following reasons. Ports in this mode will:

- Receive and process Normal Network frames as Normal Network frames (i.e., frames that are not DSA Tagged nor Ether type DSA Tagged).
- Receive and process Ether type DSA Tagged frames as DSA Tagged frames (so the CPU can control the switch).
- Can transmit Normal Network frames to the CPU as Normal Network frames or as Forward Ether type DSA Tagged frames (by selection of the port's EgressMode bits – Port offset 0x04).
- Will transmit all DSA control frames (all non-Forward DSA Tag types) to the CPU as Ether type DSA Tagged.

Note

If the CPU needs source port information on Forward DSA Tag frames then these frames cannot be sent to the CPU as Normal Network frames (i.e., the port must use EgressMode = 0x3, Port offset 0x04).

The proper usage is to set the device port used to connect to the management CPU to Ether Type DSA Tag mode (i.e., FrameMode = 0x3 at Port offset 0x04) and set the ports used to interconnect devices to each other in regular DSA Tag mode (i.e., FrameMode = 0x1) while the external ports are set to either Normal Network mode (Section 5) or Provider mode (Section 6). All FrameMode port types can co-exist as the same time on different ports of device. The format of the frames will be converted as needed.

Figure 47: Ether Type DSA Tag Format

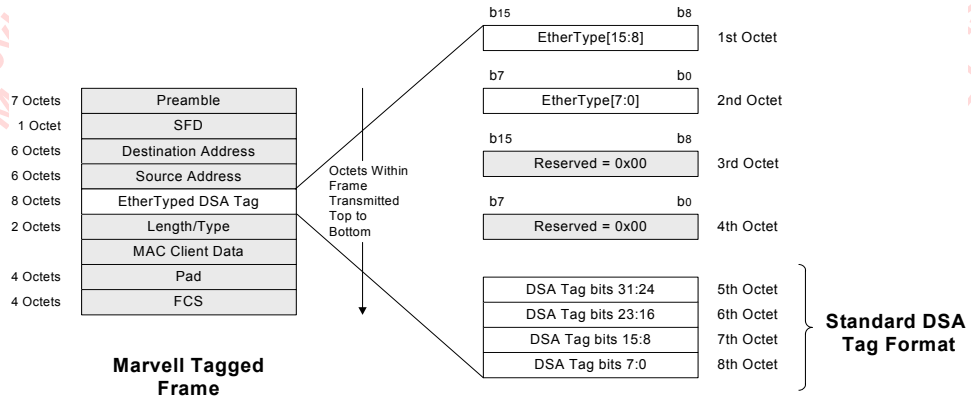


Table 39: From_CPU DSA Tag Fields

Frame's Field	Description
Ether Type	Ether type of the DSA Tag. This is a programmable value. A frame will be considered Ether type DSA tagged if its Ether type (octets 1 and 2 of the Ether type DSA Tag) equals the port's PortEType register (Port offset 0x0F).
Reserved	Sub type. Octets 3 and 4 must be written as zero.
DSA Tag	Standard DSA Tag. Octets 5 to 8 can contain any of standard DSA Tag types defined above.

Section 8. Advanced Switch Functions

The discussions that follow assume the port is in any Frame Mode (Port offset 0x04) unless specified otherwise. Each item is supported in chip or across multiple chip devices.

This section covers the following topics:

- What MGMT (Management) frames are, and how frames become MGMT frames
- How MGMT frames become normal frames
- How MGMT frames are treated differently in the switch
- Spanning Tree support is used as an example for the above items
- How to properly configure the switch connection to a management CPU and how to isolate the CPU from the frames it does not need to see
- How to properly configure the switch connection to a Router including how to increase router CPU performance using the Marvell® Header
- OAM Loopback support
- Port Mirroring support
- Port Trunking support
- Device's interrupt support

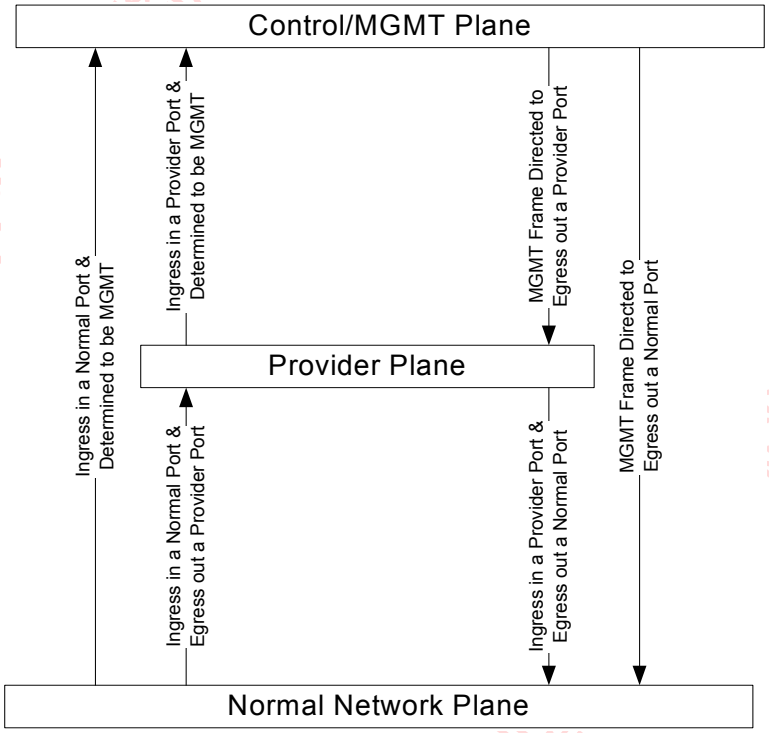
8.1 Management Frames to and from the CPU

Managed switches, those with a management CPU inside running 802.1 protocols, need to be able to detect and map special management frames to the CPU. Likewise, the management CPU needs to be able to send these special frame types out any of the switch's ports. Chip to chip management frames are needed in multi-chip designs as well, to ensure the multiple devices act as one larger switch.

The device supports a concept of a MGMT (management) data plane where control and switch management frames travel under different rules from Normal data (otherwise called Normal Network – Section 5, and/or Provider data – Section 6). MGMT frames can and do use the same physical ports as Normal frames but they are processed differently.

Figure 48 shows the concept of these separate planes and the linkage between them. The following sections cover the linkage (i.e., how to get some Normal frames to the MGMT Plane and back again) and how MGMT frames are processed inside the switch.

Figure 48: Normal and Provider vs. Control Data Planes



The handling of 802.1D's BPDU (Bridge Protocol Data Unit) frames for Spanning Tree will be used as an example throughout the discussions in this section.

8.2 Spanning Tree Support

802.1D Spanning Tree is inherently a cross-chip function as the CPU is always outside of the switch device. It is supported in the device with the help of an external CPU that runs the actual Spanning Tree algorithm. The device supports Spanning Tree by:

- Detection of BPDU[1] frames entering Network (or provider) (external switch) ports. These frames are called MGMT (management) frames in the device. They are detected by loading the BPDU's multicast address (01:80:C2:00:00:00) into the address database with a MGMT EntryState indicator (Section 9.1.1) or by using the Rsvd2Cpu bits in MGMT Enable register (Global 2 offset 0x03).
- Tunneling of BPDU frames through Blocked ports. Blocked ports are controlled by the Port's PortState bits (see Section 5.1.1 and Section 5.2.3). If a port is in the Blocked state, all frames are discarded except for frames with a DA address that is considered a MGMT address as defined in the step above.
- Redirection of BPDU frames. BPDU frames that enter a Network (or provider) port need to go to the CPU only, even though they are multicast frames. This task is handled in the BPDU frame detection phase above by mapping the BPDU's multicast address to the CPU port directly or to the port that is to be used to cascade these frames to the CPU (this is set with the value of the DPV bits when the address is loaded) or by the device's CPUDest register (Global 1 offset 0x1A). Cascaded BPDU frames egress the 1st device with a To_CPU DSA Tag (Section 7.2).
- Cascading of BPDU frames. BPDU frames that enter a DSA Tag port must enter it with a To_CPU DSA Tag. These frames are mapped directly, without modification to the device's CPUDest (Global 1 offset 0x1A). The CPUDest registers are used to form a path from all the Network ports in the switch to the CPU.
- Source Port information. The CPU needs to know the physical source port of the BPDU frame. This information is supplied in the frame's To_CPU DSA Tag (see Section 7.2) that is sent to the CPU.
- CPU transmission of BPDU frames. The CPU needs to be able to transmit BPDU frames out any physical port of the switch. This control is supported with the From_CPU DSA Tag data that the CPU needs to put on these frames before they are transmitted into the switch (Section 7.3). The Device Mapping table (Global Control 2 offset 0x06) is used to map From_CPU frames that are not destined for a local device (as marked in the DSA Tag's Trg_Dev field) out the correct port to get it to the next device. If the next device that receives the From_CPU frame is not the final destination, the process repeats until the frame gets to the target device. Once there, it will send the From_CPU frames out the target port (as marked in the DSA Tag's Trg_Port field) with the DSA Tag removed from the frame assuming the target port was a Network port.

The CPU and device hardware can support 802.1D Spanning Tree or it can be used to perform simpler bridge loop detection on a new link. The only difference is the software that runs on the attached CPU.

Notes

- For Spanning Tree to work, all device-to-device and device-to-CPU interfaces must be configured in a DSA Tag mode (FrameMode in Port Control - offset 0x04).
- Each device's CPUDest register (Global 1 offset 0x1A) must be configured pointing To_CPU DSA Tag frames toward the CPU.
- Likewise, the DeviceMapping table (Global 2 offset 0x06) must be configured to map From_CPU DSA Tag frames out the correct DSA Tag port to get the frame to the correct device if the frame is not destined for the local device.

1. BPDU = Bridge Protocol Data Unit – the frame type used to run the Spanning Tree Protocol

8.3 Ingress MGMT/BPDU Frame Detection

The device supports two methods of management/802.1D BPDU frame detection and mapping for ingressing frames from Normal Network and/or Provider ports. These two mechanisms support IEEE industry standards like STP[1], LAC[2], and OAM[3], as well as any company proprietary protocol. Both of these mechanisms:

- Detect that a frame is special by examining the frame's DA
- Assign these special frames to a category called MGMT (for management)
- Allow MGMT frames and only MGMT frames to ingress and egress Blocked ports (see Port States in Section 5.1.1)
- Set the priority on these MGMT frames overriding all other QoS decisions on the frame
- Map these MGMT frames to a port where a management CPU is directly or indirectly connected

8.3.1 Reserved Multicast Address Support

The first mechanism for MGMT frame detection is optimized to support 802.1D's 16 reserved multicast addresses. Any or all of the 16 multicast addresses in the range of 01:80:C2:00:00:0x[4] can be treated as MGMT addresses in the device. The Rsvd2Cpu register bits in the MGMT Enables 0x register (Global 2 offset 0x03) determines which of these 16 addresses are treated as MGMT and which are not as long as the Rsvd2Cpu bit in the Switch Management register (Global 2 offset 0x05) is also be set to a one.

Additionally, the 16 Generic Attribute Registration Protcol (GARP) addresses in the range of 01:80:C2:00:00:2x can be treated as MGMT addresses in the device. The Rsvd2Cpu register bits in the MGMT Enables 2x register (Global 2 offset 0x02) determines which of these 16 addresses are treated as MGMT and which are not as long as the Rsvd2Cpu bit in the Switch Management register (Global 2 offset 0x05) is also be set to a one.

Any frame, regardless of its VLAN identifier (VID) or FID[5] assigned to it, whose DA matches an enabled reserved multicast address will be considered a MGMT frame. It will be given the priority defined in the MGMT_Pri bits (Global 2 offset 0x05) and mapped to the port defined by the global CPUDest register (Global 1 offset 0x1A).

8.3.2 New and Proprietary Protocol Support

The second mechanism for MGMT frame detection is optimized to support any new, or yet to be defined, standard and/or proprietary DA based protocol. It can also be used to map any of the 32 reserved multicast addresses defined above, where the VID of the frame must be considered in the MGMT determination[6]. Any address, multicast or unicast, can be treated as a MGMT address in the device in this way. The Address Translation Unit (ATU) is used in this case. The required MAC address must be loaded into the ATU with a MGMT EntryState value, the required priority for the frame and where the frame is to be mapped (see Section 9.1.1).

Any frame whose DA matches an ATU entry with a MGMT EntryState will be considered a MGMT frame. It will be given the priority defined in the ATU's entry and mapped[7] to the port or ports defined by the entry's Destination Port Vector (DPV). The ATU supports multiple address databases (Section 4.4.8) so the DA must appear in the Forwarding Information Database (FID) assigned to the frame for the frame to be considered a MGMT frame. This feature allows a DA to be considered a MGMT address in some address databases and not in others. But it also

1. Spanning Tree Protocol
2. Link Aggregation Control
3. Operational, Administration, and Maintenance
4. Frames with a DA of 01:80:C2:00:00:01 are always treated as Pause frames and are discarded and never mapped.
5. FID is a Forwarding Information database number assigned to each frame to support multiple address databases (see Section 4.4.8).
6. If the second mechanism is being used to map any of the 32 reserved multicast addresses the bit that corresponds to the required address in the Rsvd2CpuEnables must be cleared to a zero since the first mechanism takes priority over the second mechanism.
7. EntryState = 0xE should be used so the ATU entry can force the MGMT FPri to 0x7 and QPri to 0x3 by setting its P bits to 0x7.

requires that the DA be loaded multiple times, once for each address database that needs to use this address as a MGMT address.

Note

Frames that are considered MGMT by their DA are considered MGMT in the Ingress section. That means that the frame cannot be filtered or have it mapping or priority modified due to any Normal Network Policy (Section 5) with the exception of Ingress Rate Limiting of MGMT frames (Section 5.5) or DropOnUnlock (Section 5.1.2.3) which will discard MGMT frames based on SA. The DropOnUnLock feature is not intended for DSA Ports. Instead it is intended for Normal Network or Provider Ports where frames can become MGMT as soon as they enter the switch port based on the frame's DA. Supporting DropOnUnlock in this case can help prevent BPDU (or any MGMT) DoS Attacks by discarding all MGMT frames from a particular source (or sources).

8.4 Other Ingress MGMT Frame Detection

Most 802.1 protocols require special frames to get to the management CPU. These are handled by looking at the frame's Destination Address (see Section 8.3). There are other ways to move a Normal Network (or Provider) Plane frame up to the MGMT Plane (as shown in Figure 48). These are:

- ARP Mirror frames (Section 5.3.3)
- IGMP/MLD Snooped (or Trapped) frames (Section 5.3.4).
- Layer 2 Policy Traps or Policy Mirrors (Section 5.1.3)

Note

Frames that are considered MGMT by other than the frame's DA are not considered MGMT in the Ingress section (by they are considered MGMT in egress and then on until they reach their destination). This means that these frames can be discarded due to any Normal Network Policy (Section 5). This ensures that APRs, IGMP frames, etc., are members of a port's VLAN before they will be sent to the CPU as MGMT.

8.5 MGMT Frames to Normal or Provider Egress

Figure 48 shows MGMT frames going back down to the Normal Network Plane or to the Provider Plane. This is because 802.1 protocols require that the CPU transmit special frames out the ports. If the target egress port is a Normal Network port (Section 5) or a Provider port (Section 6) then the frames need to look like normal IEEE frames.

The CPU uses From_CPU DSA Tag frames (Section 7.3) to get the special frames out a specific port. The egress logic will automatically convert the From_CPU DSA Tag to a normal IEEE frame format if the egress port is configured in Normal Network or Provider mode (FrameMode, Port offset 0x04).

Note

From_CPU DSA Tag frames are considered MGMT by the switch until the frame egresses the target port, i.e., they will egress Blocked ports (Section 5.1.1).

8.6 Proper Connection to a Management CPU

Management CPUs can easily be isolated by a 'barrier' such that they receive only the frames that they need saving their processing power for other functions. A barrier can be configured by many different ways but the easiest is clearing the EgressFloods bits to 0x0 (Port offset 0x04) on the port connected to the CPU. This will prevent all frames with an unknown Destination Address (DA), both unicast and multicast, from getting to the CPU.

Desired DA's, like the CPU's own MAC address, can be loaded into the ATU as static (Section 9.1.3 – the CPU's port will have to be a member of the frame's VLAN too, unless the CPU's MAC address is loaded as MGMT, but then it will tunnel through Blocked ports). ARP's can be mirrored (Section 5.3.3) and IGMP/MLD frames can be trapped (Section 5.3.4). Other 802.1 protocol frames can be gotten to the CPU based on the frame's DA (Section 8.3).

Other barrier options to isolate the CPU are to use Port Based VLANs (Section 5.2.1) or 802.1Q VLANs (Section 5.2.2 – where the DefaultVID on all ports, Port offset 0x07, is defined in the VTU).

The CPU can be protected by Egress Rate Shaping using frame's per second (Section 5.8.5.1). Ingress Rate Limiting (Section 5.5) on all the Normal Network and Provider Ports to limit the number of MGMT frames and/or ARP frames that are allowed to enter the switch can also be used to protect the CPU.

8.7 Proper Connection to a Router

Routers can be a CPU or some other switching device connected to a port that performs routing or higher layer switching. In this case the router needs to get all the frames so that they can be processed. 'Barriers' between some ports need to be set up to ensure frames get processed before egressing other ports on the switch. While at the same time normal switching may be allowed between some ports on the device depending upon which LAN each port belongs to. Section 5.2.1.2 gives a couple of examples on how to set up these 'barriers' using Port Based VLANs.

Usage of the Address Database needs to be considered. It may be advantageous to disable learning on some ports (LearnDisable in Port offset 0x06) where switching never occurs (like the WAN to CPU and the CPU to WAN paths). Other situations may need multiple and separate Address Databases (Section 4.4.8) and keep learning enabled on the ports. Forcing the frames to the Router may need to be done by ignoring the results of the Destination Address search (Section 8.8).

The performance of CPU routing can be greatly increased by using the Marvell® Header on the CPU's port because it aligns the IP portion of the Ethernet frame to 32-bit boundaries in the CPU's memory. The Header is also needed to limit were frames go (for single chip switch systems – in multi chip systems the same effect may be accomplished by using Cross-chip Port Based VLANs – Section 7.5.3 where the CPU can limit were frames go based upon the source port information it puts in the Forward DSA Tag). The Marvell Header (Section 8.7.1 and Section 8.7.2) can be used by itself or in conjunction with the CPU's port being in the DSA (Section 7) or Ether type DSA (Section 7.9) frame mode.

8.7.1 Switch Ingress Header for CPU Routers

The CPU in a router needs to perform many functions. One of those functions is to route IP frames from a WAN to or from LAN ports and another is to bridge frames between one VLAN and another VLAN. The device's ingress Header mode increases the performance of both of these functions. Any port can be configured to support an ingress Header by setting the port's Header bit[1] to a one (Port offset 0x04) but only the port directly connected to a CPU should be configured in this way.

The ingress Header accelerates the CPU's performance when routing IP frames by aligning the IP portion of the frame to 32-bit boundaries. This is accomplished by prepending the frame with two extra bytes of data. Bridging

1. The Header bit enables the Marvell Header mode for both ingress and egress.

between VLAN ports sometimes requires the switch to support multiple address databases (one for each VLAN) so that the same MAC address can be used on multiple VLANs. Since the CPU is generally a member of all VLANs, it must inform the switch which VLAN to use on a given frame (and thus which address database to use). This is accomplished by using an Ingress Header with a non-zero value as defined in Figure 49[1]. When the ingress Header is seen with a non-zero value its contents are written to the port's Port Based VLAN Map register (Port offset 0x06) prior to the start of the rest of the frame. The frame is then processed by the switch using this new information. In this way, the CPU can direct which port based VLAN and address database to use on every frame at wire speed.

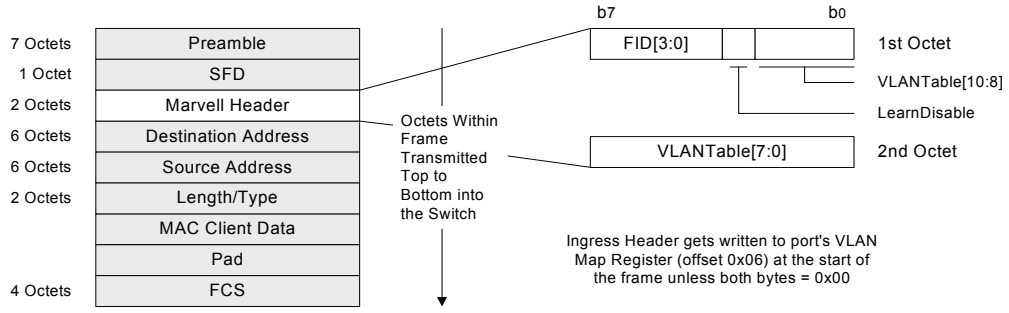
When the ingress Header mode is enabled on a port, the first two bytes of the frame (just before the DA) are used to control the switch. The Ingress Policy block removes the Header from the frame, causing the frame to be two bytes smaller in size, and overwrites the frame's FCS with a new FCS. This adjustment makes the frame normal for the rest of the network since the Header's data is intended for the switch only. Frames are padded up to 64 bytes if the size of the frame is too small after the Header is removed. This means the CPU can send 64 byte frames to the switch that contain the Header, and the frame will be accepted and resized by the switch.

The ingress Header gives the CPU the ability to control which VLAN (port based), learning mode and address database (lower 4 bits of the FID) to use on the frame that it just sent into the switch. If the CPU wants the switch to process the frame based upon the switch's current Port Based VLAN Map register settings then the CPU sets the Header data in the frame to all zeros (i.e., it prepends the frame with two extra bytes of zeros). This zero padding indicates that the switch should ignore the Header's data and process the frame normally using the current setting in the CPU port's Port Based VLAN Map register (after the Header's data is removed from the frame).

Note

Any (switch) port configured in Header mode must not receive Pause based Flow Control frames, unless the CPU places a non-zero Header in every frame it sends into the switch (i.e., Pause based Flow Control frames cannot be generated by the CPU's MAC if a Header of 0x0000 is ever used by the CPU). This is due to the fact that the ingressing Pause frame will cause the port's VLANMap register (Port offset 0x06) to be modified (MAC to MAC Pause frames will work correctly, if enabled, even if the port is in Header mode as Pause frames do not need Headers).

Figure 49: Ingress Header Format



IEEE 802.3 Frame with Marvell Header In Front of DA

Marvell Ingress Header Format

When an ingress Header contains a non-zero value its contents are written directly to the port's Port Based VLAN Map Register (Port offset 0x06) before the frame is processed. See the description of this register in the Register section (Section 13) for a description of each of the Header's fields.

1. Reserved bits in the Marvell® Header must always be zeros.

Note

The Marvell® Header can only modify the lower 4-bits of the port's FID. The upper 8-bits come from the port's FID[11:4] register bits from Port Control 1 register, Port offset 0x05.

The Marvell Header can be used with or without the DSA Tag (Section 7). In this mode, the Marvell Header's FID[3:0] is used as the port's default FID[3:0] which may be overridden by the DSA Tag's VID lookup into the VTU (If the Tag's VID is contained in the VTU). The Marvell Header can be used in single-chip or cascaded chip environments although some of its usefulness is reduced in a cascaded environment owing to the limited VLANTable size in the Header (but Cross-chip Port Based VLANs can be used instead – Section 7.5.3).

The DSA Tag can be used to force where a frame goes (by using the From_CPU format). The Marvell Header is used to limit where a frame goes (by using the VLANTable).

8.7.2 Switch Egress Header for CPU Routers

If a CPU wants to have the IP frame data of the Ethernet frame aligned to 32-bit boundaries for faster routing, the device supports a Marvell Header Mode that inserts two bytes into the frame just before the frame's Destination Address (DA). Any port can be configured this way by setting the Header bit to a one (Port offset 0x04) but only the CPU's port should be configured this way.

When the egress Header mode is enabled on a port, two extra bytes are added to the beginning of the frame just before the frame's DA and a new CRC is calculated for the frame. When the frame is received by the CPU the Header will be the first two bytes of the frame in memory. If the CPU's MAC needs to process the frame for filtering or for other reasons, the MAC must be aware that the frame data has been shifted down by two bytes. If the routing CPU places its MAC in promiscuous mode, the shifting of the frame's data will not have any effect.

The Egress Header increases routing performance since the frame can be routed 'in place' in the CPU's memory without needing to copy it to get the IP portions of the frame on 32-bit boundaries (and then copy it again before transmission).

The format of the egress Header is shown in Figure 50 and its fields are defined in Table 40.

Figure 50: Egress Header Format

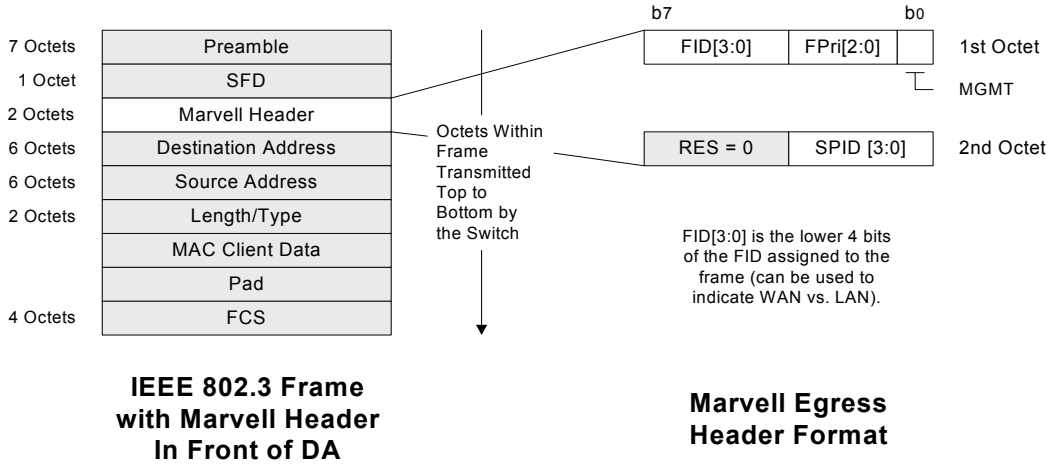


Table 40: Egress Header Fields

Frame's Field	Description
FID[3:0]	Forwarding Information Database number. This field represents the lower 4 bits of the address database number assigned to this frame when it ingressed into this device (i.e., it is assigned by the last device this frame entered). It can be used to indicate the logical port based VLAN number of the source port.
FPri[2:0]	The frame's priority as determined by the ingress rules of the last device this frame entered (see Section 5.4). If the frame is a MGMT frame (see Section 8.1), the ingress rules effectively make this the frame's priority as determined by the ingress rules of the 1st or original device this frame entered.
MGMT	Management. This bit is set to a one if the frame is a MGMT frame (see Section 8.1).
RES	Reserved for future use. Currently set to 0x0.
SPID[3:0]	The Source Port ID. These bits indicate at which physical port the frame entered this device (i.e., it is assigned by the last physical device this frame entered). An SPID of all zeros indicates Port 0. An SPID of 0x1 indicates Port 1. 0x2 indicates Port 2, 0x03 indicates Port 3, etc.

8.8 MUX'ing or Ignoring Address Translation

The device supports the ability to ignore the results of a frame's DA lookup in the Address Translation Unit. The use of the DA mapping results can be enabled or disabled on a port-by-port basis by changing the value of the port's MapDA bit (Port offset 0x08). DA lookups always take place, regardless of the setting of port's MapDA bit, however. This is done so that DA lookups that are found to be MGMT entries in the ATU (Section 8.3.2) are always mapped (even if the MapDA bit is configured to ignore the results). The application of all other switch policies are not effected by the port's MapDA bit.

When DA mapping is disabled (the port's MapDA bit equals zero) all non-MGMT frames that enter the port will be mapped based on the VLAN rules that are applied to the frame (Section 5.2) along with the Egress Flooding rules (Section 5.8.1 and Section 5.8.2). These bits can be configured in such a way that all non-MGMT frames that enter a port egress out a specific port (even the frame's source port, Section 5.3.1). This can be used for the following applications:

8.8.1 Passing Frames to a Router

The MapDA bit can be used to MUX all non-MGMT frames that enter a port to go out another port where a router can perform more processing on the frame. VLANs are used to get the frame to the router's port (Section 5.2). MGMT frames will be mapped as they normally would (Section 8.3).

The router may decide the frame can go where it normally would have gone and returns the frame back to the switch unmodified. Assuming the ingress port on which the router is attached to has its MapDA bit set (i.e., enabled), the frame will now be mapped using the address database information. If the frame's DA is unknown or is a multicast address, the frame will flood out all the ports of the frame's VLAN including the port the frame originally came in on if the port is also a member of the frame's VLAN. Use a different VID on the frame to prevent this.

Alternatively, use Distributed Switch Architecture (DSA) Tags on the port connected to the router (Section 7) and enable the global LoopbackFilter bit (Global 2 offset 0x05). DSA Tags contain the original source port information in the frame. So when the router sends the frame back into the switch (with the DSA Tag portion of the frame unmodified) the switch knows to prevent sending the frame back out its original source port (when LoopbackFilter is set to a one).

8.8.2 Operational, Administration, and Maintenance (OAM) Loopback

The MapDA bit can be used to MUX all non-MGMT (i.e., non-OAM) frames that enter a port to go back out the port they came in on. Port Based VLANs are used to get the frame to go back out the original port and only the original port (Section 5.3.1). MGMT frames, including OAM control frames, will be mapped as they normally would (Section 8.3).

Note

The best way to loop non-MGMT frames back out the port they came in on is by using the VLANTable (Port offset 0x06). Set the source port's bits only in the VLANTable. Address learning can be disabled during loopback, if desired, by setting the source port's LearnDIsable bit to a one (Port offset 0x06).

8.9 Port Mirroring Support

Port mirroring or monitoring is supported by the device with Egress only monitoring, Ingress only monitoring or Egress and Ingress monitoring. Egress monitoring sends any data that egresses out a particular port to a specific monitor port as well. Ingress monitoring sends any good data that ingresses in a particular port out to a specific monitor port as well as sending the frame where it normally would have gone.

This form of port monitoring is enabled by defining which port or ports are to be the Ingress Monitor Source (IMS) and/or the Egress Monitor Source (EMS). A port becomes an IMS and/or an EMS by setting the appropriate bit(s) to a one in the port's Port Control 2 register (Port offset 0x08). Both of these bits can be set at same time on the same port and multiple ports can have their bits set.

While many ports can be defined to be the monitor source only one destination port for the IMS frames and one destination port for the EMS frames per devices can be defined. Frames that are received on Ingress Monitor Source ports (IMS) are copied to the port defined as the Ingress Monitor Destination (IMD). Frames that are transmitted out Egress Monitor Source ports (EMS) are copied to the port defined as the Egress Monitor Destination (EMD). The IMD and EMD are defined in the Monitor Control register (global offset 0x1A). The IMD and EMD can point to the same physical port.

Each device that has at least one monitor source port enabled must also have a monitor destination port defined (of the same type - i.e., an IMD for an IMS and an EMD for an EMS). If the destination port is a Normal Network port connecting to the outside world, the frame will be copied there and that is the end of it. If the destination port is a DSA Tag port being used to connect to another device, the frame will be copied there, but the frame will egress with a To_Sniffer DSA Tag and the tag will indicate if the frame is from an IMS or an EMS.

The device that receives the To_Sniffer DSA Tag will map those frames to that device's IMD or EMD depending upon the indication in the DSA Tag frame. These frames are not learned from and they are not filtered in any way. They simply progress to the appropriate monitor destination port. If that destination port is another DSA Tag port, the frame egresses unmodified with its original To_Sniffer DSA Tag and the process continues. If the destination port is a Normal Network port, the To_Sniffer DSA Tag is removed and the frame egresses looking as it originally looked at the monitor source port.

Cross-chip port monitoring requires the following:

- Any time a monitor source is enabled in a device, the associated monitor destination must also be defined.
- A monitor source cannot be a DSA Tag port unless it is the CPU's port. Basically it is best if the monitor source ports are Normal Network ports and final monitor destination ports are also Normal Network ports.
- If the final monitor destination is cross-chip in another device, the monitor destinations in each device must form a complete path toward the final monitor destination port using DSA Tag enabled ports on both sides of the connections.
- All final Network monitor destination ports (both EMD and EMD) must be isolated from all the other local ports in the switch to prevent these ports from getting flooding frames from non-monitor source ports. The best way to isolate these ports is to use port based VLANs (Section 5.2.1).
- The final egress monitor destination port's 802.1Q Mode and VID MemberTag information must be configured to match the egress monitor's source port's configuration. If this is not done, the frame will still egress the final EMD port with the correct data in the frame but its tag mode (i.e., egressing tagged or egressing untagged) may not match the way the frame egressed the original EMS port.

In the device the following frames that ingress a port are not forwarded for Monitoring:

- Frames received by the MAC but were not stored by the switch due to a lack of memory
- Frames received with a bad CRC (unless the CRC is fixed by the ForceFCS bit being set on the port, Port offset 0x08)
- Pause frames that are received
- Frames with a size less than 64 bytes or greater than the maximum size allowed

- Frames discarded due to Ingress Rate Limiting (Section 5.5)
- Frames received but sampled[1] due to Ingress Rate Limiting (88E6097 only – Section 5.5)

The following frames that ingress a port are forwarded only for Monitoring (i.e., they are not sent to any other port):

- Frames that are discarded for 802.1X Source MAC address security
- Frames that are discarded for 802.1Q security
- Tagged frames received when DiscardTagged is enabled
- Untagged frames received when DiscardUntagged is enabled
- Frames that are discarded due to the port's PortState setting
- Frames that aren't mapped to any other port due to DA mapping and VLAN restriction

1. One of the port's Ingress Rate Limiting resources can be used to mirror 1 of N received IMS frames from the port (see Section 5.5).

8.10 Port Trunking Support

Port trunking is supported by the device with any combinations of ports (in-chip and cross-chip). The ports that are to be associated with the trunk need to have all the port member's defined with the same TrunkID (Port offset 0x05) and have their Trunk Port bit set to a one (also at Port offset 0x05). Up to 16 trunk groups are supported with up to eight ports per trunk group.

8.10.1 Trunk Address Learning

When a frame enters a Trunk Port its Source Address (SA) is learned (or loaded) with its association being to the ingress port's TrunkID number (its T bit will be a one - Section 9.1.1). This way the contents of the address database contain the same association with the frame's SA regardless of what link of the trunk the frame entered the switch. If this frame egresses a DSA Port (Section 7) it will be marked as coming from a trunk port and its Src_Port/Src_Trunk field will contain the TrunkID of the first trunk port the frame entered. This ensures that all devices in the switch learn this frame's SA with the source port's TrunkID.

8.10.2 Trunk Address Searching

When frames are destined back toward a trunk the frame will have its Destination Address (DA) searched for in the address database. If the frame's DA is unknown the frame will try to flood out all ports of the trunk (this is OK so far as this will be fixed with load balancing). If the frame's DA is found the entry will indicate the entry is mapped to a trunk (its T bit will be a one - Section 9.1.1) and the entry's DPV bits will contain the TrunkID associated with the frame's DA. This TrunkID needs to be converted into a DPV (Destination Port Vector) the rest of the switch can use. This is accomplished by accessing the Trunk Mapping table (Global 2 offset 0x08) using the TrunkID that was in the ATU's entry.

8.10.3 Trunk Mapping

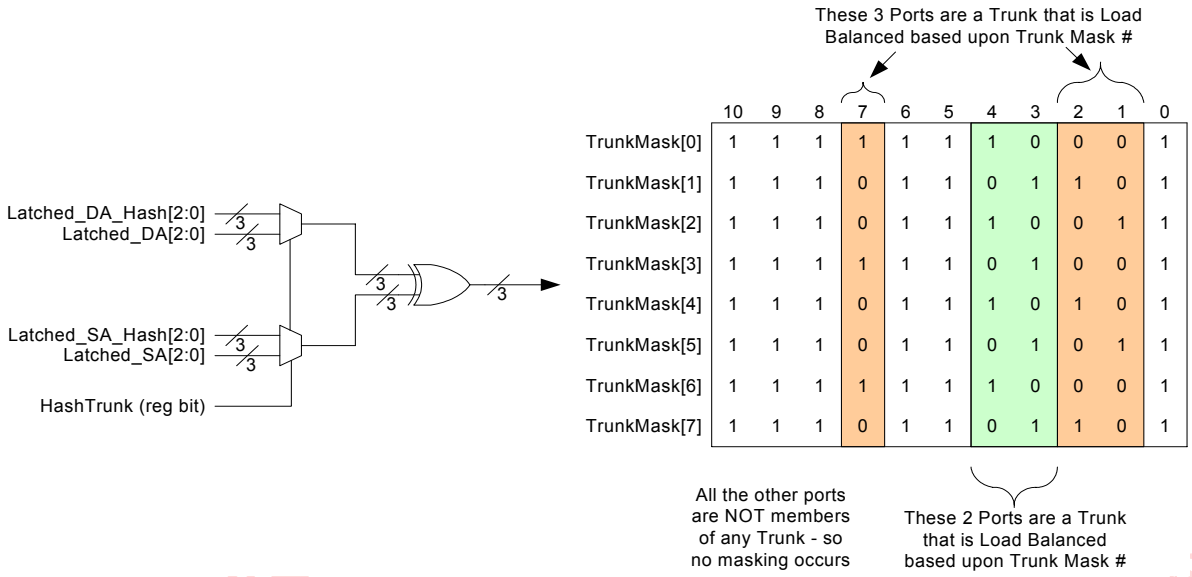
The Trunk Mapping table (Global 2 offset 0x08) needs to be configured in each device in the switch by software for each TrunkID number that is in use. It is used to convert found DA searches that return an entry that is associated with a TrunkID (one with its T bit set to a one - Section 9.1.1) into a DPV (Destination Port Vector). Software configures each TrunkID entry in the table by placing a one in the entry for every port on the local device that is a direct or indirect member of the Trunk. Direct members of a trunk are ports with their Trunk Port bit set to a one (Port offset 0x05) with a TrunkID (also at Port offset 0x05) value equal to the Trunk Mapping table's entry being modified. Indirect members of a trunk are local device DSA Ports (Section 7) that connect to another switch device that contains direct members of the same TrunkID. The multiple bits being set in the resulting DPV ensures the frame is mapped to all possible members of the trunk (this is OK so far as this will be fixed with load balancing).

8.10.4 Load Balancing

Load balancing is used to ensure frames only egress one link (or port) member of a trunk. DA/SA based load balancing is used by configuring the Trunk Mask table (Global 2 offset 0x07). Software must configure all eight entries for all ports in any device that contains direct members of a trunk (Section 8.10.3). Load balancing will not occur on MGMT frames (Section 8.1).

The Trunk Mask table (Figure 51) contains a bit per port for eight mask or load balance settings. Each frame that ingresses the switch selects one of the eight possible masks (depending upon the frame's DA and SA) and the selected mask is used to ensure each frame egresses only one port on each trunk.

Figure 51: Trunk Mask Load Balancing Table - Example



The lower 3-bits of the frame's DA and SA are XOR'd together. The resulting 3-bit value used to select one of the 8 trunk masks. If the lower 3-bits of a frame's DA or SA changes the selected trunk mask will be different. The device supports an alternate DA/SA mapping into the table. Instead of using the lower 3-bits of the frame's DA and SA, the lower 3-bits of a Hash function applied to the DA and SA can be used. This alternate hash approach randomizes the load balancing, while the original direct DA/SA approach is easier to test and verify that the Trunk Mask table is configured correctly. The selected Trunk Mask value is used to prevent frames from egressing ports they would normally egress based upon the frame's DA and SA. The frame will not egress out a port if the selected Trunk Mask bit for that port is a zero.

The reset values in the Trunk Mask table is all one's for all ports for all entries. This implies that none of the ports in this device are a member of a trunk since no load balancing is performed on any of the links (i.e., all frames are allowed to egress all ports regardless of the frame's SA and DA).

Figure 51 shows an example where ports 3 and 4 form one trunk and ports 1, 2 and 7 form a three port trunk. Ports that are not members of any trunk must have their Trunk Mask bits set to a one in all Trunk Mask entries. This keeps normal data flowing out those ports regardless of the frame's DA and SA.

Ports that are members of a trunk must have one and only one of the trunked port's Trunk Mask bit set to a one for each Trunk Mask entry. The other members of the trunk must have their Trunk Mask bit cleared to a zero. The single trunk member (port) that has it's Trunk Mast bit set to a one will be the one trunk link that will egress frames with an DA/SA combination that selected that Trunk Mask entry. This 'steers' the frame out one, and only one, port of the trunk based upon the frame's DA and SA. All eight Trunk Mask table entries must be configured as all eight entries will be used based upon the DA and SA of frames going through the switch. The eight Trunk Mask table entries support trunks up to eight ports is size.

Software can move a flow from one port to another port in the trunk by changing the values in the Trunk Mask table as long as the rules above are followed (i.e., no trunk has more than one 'one' in each Trunk Mask table entry).

8.11 Interrupt Controller

The device contains an Interrupt Controller used to merge various interrupts into the device's INTn pin. The Switch Global Control register (Global 1 offset 0x04) determines whether or not the INTn pin is asserted when an interrupt occurs. Each interrupt bit in the Switch Global Control register (DeviceInt, StatsDone, VTUProb, VTUDone, ATUProb, ATUDone, PHYIntEn, or EEINTn) can be individually unmasked to enable a switch core interrupt. PHY interrupts are enabled through the PHYIntEn bit, which is in the Switch Global Control register as well.

The Switch Global Status Register (Global 1 offset 0x00) reports the interrupt status. When an unmasked interrupt occurs and the INTn asserts, the CPU needs to read the Switch Global Status register to determine the source of the interrupt.

- The EEInt indicates the processing of the EEPROM contents is complete and the I/O registers are now available for CPU access. A CPU can use this interrupt to know it is OK to start accessing the device's registers. The EEInt will assert the device's INT pin even if no EEPROM is attached unless the EEPROM changes the contents of the EEIntMast register (Global 1, offset 0x04) or if the Test SW_MODE has been configured (Table 17).
- The StatsDone, VTUDone and ATUDone interrupts de-assert after the Switch Global Status register is read (these interrupt status bits are clear on read). These interrupts indicate that the last Stats operation, VTU operation and/or ATU operation has completed.
- The PHYInt indicates that one or more of the PHY interrupts enabled in PHY register offset 0x12 are active. The PHYInt will stay low until the PHY interrupts are serviced (see the PHY functional description on how to process its interrupts – See PHYInt bit Table 96).
- The DeviceInt indicates that the source of the interrupt is from the Interrupt Source register (Global 2 offset 0x00). The DeviceInt will stay low until the source of the interrupt is serviced.

8.11.1 Device Interrupts

Each interrupt supported in the Interrupt Source register (Global 2 offset 0x00) can be independently masked by a bit in the Interrupt Mask register (Global 2 offset 0x01). All the bits in the Interrupt Source register are clear on read and they can indicate:

- A WatchDog event occurred. WatchDog events are enabled in Global 2 offset 0x1B.
- A Jam Limit event occurred (Section 4.3.6).
- Link status changed on one of the SERDES interfaces (ports 10:8).

Note

The active level of the fiber signal detect pin (SDET) can be inverted on a port by port basis for the SERDES ports and the 10/100 ports (if the PHY is in fiber mode). See the SDET Polarity register (Global 2, offset 0x1D).

Section 9. Accessing Data Structures

The device contains many data structures that are used to control switching. The larger structures have specialized ways to access them and they are capable of generating an interrupt to the CPU if they need servicing. These larger structures are the Address Database controlled by the ATU (Section 9.1) and the VLAN Databases controlled by the VTU (Section 9.2).

9.1 Address Translation Unit Operations

The Address Translation Unit (ATU) in the device supports user commands to access the contents of the MAC address database.

All ATU operations have the same user interface and protocol. Six global registers are used and are shown in Table 41. The protocol for an ATU operation is as follows:

- Ensure that the ATU is available by checking the ATUBusy bit in the ATU Operation register. The ATU can only perform one user command at a time.
- Load the ATU Data, ATU FID and ATU MAC registers if required by the selected operation.
- Start the ATU operation by defining the desired ATUOp and setting the ATUBusy bit to a one in the ATU Operation register – this can all be done at the same time.
- Wait for the ATU operation to complete. This is done by polling the ATUBusy bit in the ATU Operation register or by receiving an ATUDone interrupt (see Switch Global Control, Global 1 offset 0x04, and Global Status, Global 1 offset 0x00).
- Read the results if appropriate.

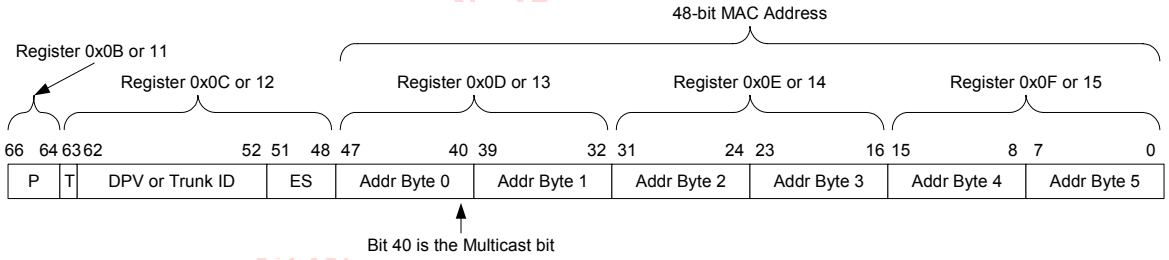
Table 41: ATU Operation Registers

Register	Offset	Before the Operation Starts	After the Operation Completes
ATU FID	0x01 (decimal 1)	Used to define which address database to use.	Used to indicate the returned MAC's FID.
ATU Operation	0x0B (decimal 11)	Used to define the MAC's Priority, the required operation and start the ATU operation.	Used to indicate the ATU's Busy status and the returned MAC Priority and.
ATU Data	0x0C (decimal 12)	Used further to define the required operation and used as the required ATU Data that is to be associated with the MAC address below.	Returns the ATU Data that is associated with the resulting MAC address below.
ATU MAC (3 registers)	0x0D to 0x0F (decimal 13 to 15)	Used to define the required MAC address upon which to operate.	Returns the resulting MAC address from the desired operation.

9.1.1 Format of the ATU Database

Each MAC address entry in the ATU database is 67-bits in size. The lower 48 bits contains the 48-bit MAC address and the upper 19 bits contains information about the entry as shown in Figure 52. The database is accessed 16-bits at a time via the Switch Global 1 registers shown in Table 42.

Figure 52: Format of an ATU Entry



The right 48-bits in Figure 52 is the 48-bit MAC address associated with this ATU entry. The upper 19 bits is the data that is associated with the entry's MAC address. The upper 19 bits are defined as follows:

Table 42: ATU Data bits

Field	Bits	Description
P	66:64	The MAC's Priority override value when enabled by the EntryState bits below. Used for priority override on ingressing frames (see Section 5.4.6). Enabling a priority on a MGMT MAC address (Multicast EntryState = 0xE or Unicast EntryState = 0xD) will override *all* priorities for these MGMT frames. Enabling a priority on a static, non-MGMT MAC address, will only override the frame's priority if the port's DAPriOverride and/or SAPriOverride bits are configured to do so (Port offset 0x0C).
T	63	The Trunk bit used to qualify the contents of the DPV or Trunk ID bits below. When this bit is zero bits 62:52 are the DPV (Destination Port Vector) associated with this MAC. When this bit is a one, bits 55:52 is the Trunk ID associated with this MAC (bits 62:56 must be zero in this case).
DPV or Trunk ID	62:52	The Destination Port Vector or Trunk ID. When the Trunk bit (bit 63 above) is a zero these bits indicate which port or ports are associated with this MAC address (i.e., where frames should be switched) when they are set to a one. A DPV of all zeros indicates frames with this DA should be discarded and/or special handling of frames with this SA should occur (see Section 4.4.7 and Section 5.1.2). Bit 52 is assigned to physical Port 0, 53 to Port 1, 54 to Port 2, etc. If more than one port's bit is set to a one frames mapped to this MAC address will attempt to egress out more than one port. This is used for multicast filtering. When the Trunk bit (bit 63 above) is a one bits 55:52 (the lower 4 bits) indicate the Trunk ID that is associated with this MAC address (in this case bits 62:56 must be zeros). The port or ports that this DA MAC address are mapped to is determined by the contents of the Trunk Mapping Table (Global 2 offset 0x08).

Table 42: ATU Data bits (Continued)

Field	Bits	Description
EntryState	51:48	The EntryState field, together with the entry's Multicast bit (bit 40) is used to determine the entry's age or its type as follows: For unicast MAC addresses (bit 40 = 0): • 0x0: Unused entry • 0x1 to 0x7: Used entry where EntryState = the Age of the entry where 0x1 is the oldest • 0x8: Static Policy entry (88E6097 only) • 0x9: Static Policy entry with Priority Override (88E6097 only) • 0xA: Static Non Rate Limiting (NRL) entry • 0xB: Static Non Rate Limiting (NRL) entry with Priority Override • 0xC: Static entry defining frames with this DA as MGMT • 0xD: Static entry defining frames with this DA as MGMT with Priority Override • 0xE: Static entry • 0xF: Static entry with Priority Override For multicast MAC addresses (bit 40 = 1): • 0x0: Unused entry • 0x1 to 0x3: Reserved for future use • 0x4: Static Policy entry (88E6097 only) • 0x5: Static Non Rate Limiting (NRL) entry • 0x6: Static entry defining frames with this DA as MGMT • 0x7: Static entry • 0x8 to 0xB: Reserved for future use • 0xC: Static Policy entry with Priority Override (88E6097 only) • 0xD: Static Non Rate Limiting (NRL) entry with Priority Override • 0xE: Static entry defining frames with this DA as MGMT with Priority Override • 0xF: Static entry with Priority Override Auto learned entries (Section 4.4.3) will always be unicast entries with an EntryState value in the range of 0x1 (oldest age) to 0x7 (recently added or refreshed). Usage of the various other kinds of EntryState values are covered in the following sections: • Policy EntryState values is covered in Section 5.1.3 • Non Rate Limiting EntryState values is covered in Section 5.5 • MGMT (management) EntryState values is covered in Section 8.3.2 • Priority Override EntryState values is covered in Section 5.4.5

9.1.2 Reading the Address Database

The contents of the address database can be dumped using Ethernet frames that get transmitted to the CPU (or other device). Up to 48 valid entries can be retrieved in each frame. See Remote Management described in Section 10.

Alternatively, the contents of the address database can be dumped or searched using the register interface. The dump operation is called Get Next since it returns the active contents of address database in ascending network byte order. A search operation can also be done using the Get Next operation. If multiple address databases are being used (see Section 4.4.8), set the FID field in the ATU FID register to the database number to search when using the Get Next function.

The Get Next operation starts with the MAC address contained in the ATU MAC registers and returns the next higher active MAC address currently in the address database. Begin with an ATU MAC address of all ones to get the first or lowest active MAC address. The returned MAC address and its associated data is accessible in the ATU MAC and the ATU Data registers. To get the next higher active MAC address, the Get Next operation can be started again without setting the ATU MAC registers since they already contain the 'last' address. A returned ATU MAC address of all ones indicates that no higher active MAC addresses were found or that the Broadcast MAC address was found. In either case, the end of the database has been reached. If it were reached with a valid Broadcast address the entry's EntryState is returned with a non-zero value. A summary of how the Get Next operation uses the ATU's registers is shown in Table 43.

Table 43: ATU Get Next Operation Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
ATU FID	0x01 (decimal 1)	Used to define which address database to use.	Used to indicate the returned MAC's FID.
ATU Operation	0x0B (decimal 11)	Used to define the required operation and start the ATU operation.	Used to indicate the ATU's Busy status and report the resulting MAC's priority.
ATU Data	0x0C (decimal 12)	Ignored.	Returns the ATU Data that is associated with the resulting MAC address below. If EntryState = 0x0 the returned data is not a valid entry.
ATU MAC (3 registers)	0x0D to 0x0F (decimal 13 to 15)	Used to define the starting MAC address to search. Use an address of all ones to find the 1st or lowest MAC address. Use the last address to find the next address (there is no need to write to this register in this case).	Returns the next higher active MAC address if found, or all ones are returned indicating the end of the table has been reached (all ones is a valid entry if EntryState ≠ 0x0).

To search for a particular MAC address, start the Get Next operation with a MAC address of one less than the target MAC address using the FID of the database to search. If the required MAC address is found it is returned in the ATU MAC registers along with its associated data in the ATU Data register. If the searched MAC address is not found active, the ATU MAC registers will not equal the target address.

9.1.3 Loading & Purging an Entry in the Address Database

Any MAC address (unicast or multicast) can be loaded into, or removed from, the address database by using the Load operation. An address is loaded into the database if the EntryState in the ATU Data register is non-zero. A value of zero indicates that the ATU operation is a purge.

The Load operation searches the address database indicated by the Forwarding Information Database number (ATU FID, Global 1 offset 0x1), for the MAC address contained in the ATU MAC registers. If the address is found it is updated by the information found in the ATU Data and Operation registers.

Note

A load operation becomes a purge operation if the ATU Data's EntryState equals zero.

Note

Static addresses can be modified without first being purged.

If the address is not found, and if the ATU Data register's EntryState does not equal zero, the address is loaded into the address database using the same protocol as automatic Address Learning (see Section 4.4.3). The 16 bits of the ATU Data register are written into bits 63:48 of the ATU entry (see Section 9.1.1). The 3 MACPri bits of the ATU Operation register are written into bits 66:64 of the ATU entry.

A summary of how the Load operation uses the ATU's registers is shown in Table 44.

Table 44: ATU Load/Purge Operation Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
ATU FID	0x01 (decimal 1)	Used to define which address database to use.	No change.
ATU Operation	0x0B (decimal 11)	Used to define the operation, the priority to associate with the entry, and start the ATU Operation.	Used to indicate the ATU's Busy status.
ATU Data	0x0C (decimal 12)	Used to define the associated data that is loaded with the MAC address below. When EntryState = 0, the load becomes a purge.	No change.
ATU MAC (3 registers)	0x0D to 0x0F (decimal 13 to 15)	Used to define the MAC address to load or purge.	No change.

9.1.4 Flushing Entries

All MAC addresses or just the unlocked (Non-Static) MAC addresses can be purged from the entire set of address databases or from just a particular address database using single ATU operations. These ATU operations are:

- Flush all Entries
- Flush all Non-Static Entries
- Flush all Entries in a particular FID Database
- Flush all Non-Static Entries in a particular FID Database

The Flush requires that the EntryState bits in the ATU Data register be 0x0. The ATU MAC Address registers are not used for these operations and they are left unmodified. The FID of the ATU FID register is used for the Flush operations that require a database number to be defined.

9.1.5 Moving or Removing Single Port Mappings

All MAC address port mappings associated with a specific port can be moved to another port or removed from the address database. This operation can be carried out for the entire set of address databases or for just a particular address database using single ATU operations. These ATU operations are:

- Move all Entries
- Move all Non-Static Entries
- Move all Entries in a particular FID Database
- Move all Non-Static Entries in a particular FID Database

The Move requires the EntryState bits in the ATU Data register (Global 1 offset 0x0C) to be 0xF. The PortVec bits in the ATU Data register are used to define the FromPort (bits [3:0] of PortVec) and the ToPort (bits [7:4] of PortVec). The ATU MAC Address registers (Global 1 offsets 0x0D to 0x0F) are not used for these operations and they are left unmodified. The FID field of the ATU FID register (Global 1 offset 0x01) is used for the Flush operations that require a database number to be defined.

An ATU Move operation examines all the entries in the address database. Any valid entry that meets the requirements is processed. The requirements are:

- The address is valid (its EntryState is non-zero)
- The address is contained in the selected database (either all or the one FID selected)
- The address is not associated with at Trunk (the entry's 'T' bit is not set)
- The address has the selected state (either all or Non-Static)
- The address has the FromPort's bit set to a one in its DPV field

Processed entries have their ATU Data register FromPort bit cleared to a zero and have their ToPort bits set to a one. If the ToPort's value is 0xF, the FromPort bits are cleared but the ToPort's bit is not set. This is how entries associated with a particular port can be removed from the address database.

Note

Entries associated with a Trunk cannot be moved or removed using these commands.

Note

This mechanism moves the entries inside this device only. This feature does not work across multi-chip implementations.

9.1.6 Servicing ATU Violations

The ATU captures ATU Full, SA Member Violation, SA Miss Violation and ATU Age Out Violation data.

- An ATU Full violation occurs if an Automatic Address Learn (Section 4.4.3) or an ATU load operation (Section 9.1.3) could not enter the new MAC address into the address database owing to all four bins at the MAC address's hashed address being locked as static entries.
- An SA Membership Violation occurs when a frame's SA is found in the address database as static and the entry's Destination Port Vector (DPV) data does not align with port's number or mode[1]. It also occurs if the port is Locked (Port offset 0x0B), and if the port's RefreshLocked bit is cleared to zero (Port offset 0x0B) and the ATU Age Interrupt is not enabled (Global 2 offset 0x5) and the port's DPV data does not align with port's number or mode. If the ATU Age Interrupt is enabled and the entry's EntryState is less than 0x4 an ATU Miss Violation will be generated instead of the Member Violation. This Membership Violation interrupt can be masked on a per-port basis by setting the port's IgnoreWrongData bit to a one (Port offset 0x0B).
- An SA Miss Violation occurs when a frame's SA is not found in the address database and the port is locked due to the port's LockedPort bit being set to a one (Port offset 0x0B). This 1st learn interrupt can be masked on a per-port basis by clearing the port's LockedPort bit to zero. The SA Miss Violation will also occur if the port is locked and the frame's SA is found in the address database but its EntryState is less than 0x4 and the ATUAgeIntEn bit is set (Global 2 offset 0x5). This refresh interrupt allows the CPU to reload aging ATU entries before they age out, if they are still being used by ingressing frames. The aging interrupt will not happen if the port is configured to auto refresh already learned entries (see RefreshLocked, Port offset 0x0B).
- An ATU Age Out Violation occurs when an entry is at EntryState 0x1 and it is being aged again, and the port the ATU entry is associated with[2] has its IntOnAgeOut bit set (Port offset 0x0B). Up to two Age Out Violations can be stored and they are serviced ahead of any Full, Member or Miss Violations. With the default AgeTime of 0x16 (330 second age time – Global 1 offset 0x0A) the fastest rate two back-to-back ATU entries can age is 5.75 mSec. So the two deep FIFO will not miss any Age Out Violations unless the CPU takes more than 10 mSec to service the interrupt. For CPU directed learning, it can assure that no Age Out Violations are missed by setting the port's HoldAt1 bit (Port offset 0x0B). The HoldAt1 option ensures that the automatic aging unit never actually purges the entry by decrementing the entry's EntryState to 0x0. Instead it will hold all entries associated with the port at an EntryState of 0x1. The CPU will get another Age Out Violation from this entry on the next age sweep through the address database. When the port's HoldAt1 bit is set, the CPU has to purge all entries as they will not age out on their own.

Captured ATU Violations and their associated interrupts are cleared by the Get/Clear Violation Data ATU Operation. This ATU Operation returns the type of the violation in the ATU Operation register (Global 1 offset 0x0B), the source port that caused the violation in the EntryState/SPID field of the ATU Data register[3] (Global 1 offset 0x0C) and returns the MAC address that caused the violation in the ATUByte[5:0] fields of the ATU MAC register (Global 1 offsets 0x0D to 0x0F) and the FID that was associated with the frame (Global 1 offset 0x01).

A summary of how the Get/Clear Violation Data operation uses the ATU's registers is shown in Table 45.

1. If the port is not a Trunk port (Port offset 0x05) then the port's bit must be set in the DPV and the entry must not be a Trunk entry (i.e., its 'T' bit must not be set – Section 9.1.1) or a violation will be generated. If the port is a Trunk port then the entry must be a Trunk entry with its DPV[3:0] matching the port's Trunk ID (Port offset 0x05) or a violation will be generated.
2. An entry is associated with a port when the entry is a non-trunk entry (the entry's 'T' bit = 0) and the port's bit is set in the entry's DPV. If the found entry contains a Trunk ID, the Trunk ID is converted to a DPV using the Trunk Mapping Table (Global 2, offset 0x08) and then it is processed in the same way.
3. Except for ATU Age Out Violations

Table 45: ATU Get/Clear Violation Data Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
ATU FID	0x01 (Decimal 1)	Ignored.	Used to indicate the returned MAC's FID.
ATU Operation	0x0B (Decimal 11)	Used to define the desired operation and start it	Used to indicate the ATU's Busy status and the type of the violation.
ATU Data	0x0C (Decimal 12)	Ignored.	Used to indicate the SPID that was involved in the violation unless this is an Age Out violations where this is the entry's data instead.
ATU MAC (3 registers)	0x0D to 0x0F (Decimal 13 to 15)	Ignored.	Used to indicate the MAC that was involved in the violation

9.1.7 ATU Statistics

The ATU supports a simple set of statistics counters to help determine the current usage of the address database. An overview of the address database's structure will help with the meaning of these counters.

The address database's 8,192 entries are organized as 2,048 buckets with each bucket containing 4 bins (Section 4.4.1). The buckets are addressed by hashing the 48-bit MAC address down to 11 bits. As multiple MAC addresses can hash to the same bucket (a hash collision), 4 bins in each bucket are provided. The 1st bin is filled first, then the 2nd, and so on.

The ATU Statistics return the quantity of the selected entries that are currently present in the address database in each of the four bins. Adding the count from the four bits together gives a total entry count. The separate bin values give an indication on how many hash collisions are currently in the address database. For example: If bins 3 and 4 are empty, all addresses presented to the address database have been learned without any problems. If bin 4 is empty but bin 3 is over 1,024 entries, the address database has still learned all addresses, but it is nearing its limit. If bin 4 is non-zero, some least recently used addresses may have been overwritten by newer addresses. And if bin 4 is over 1,024 entries the database is severely stressed.

The ATU Statistics are gathered every time an ATU GetNext is performed (Section 9.1.2) and the results are held in the counters until the start of the next GetNext operation. The statistics to gather can be selected prior to the start of the ATU GetNext operation. The choices are:

- Count all valid entries
- Count all valid non-static entries (counts dynamic entries only)
- Count all valid entries in a defined FID (Section 4.4.8)
- Count all valid non-static entries in a defined FID

If the defined FID option is selected the FID counted is the one defined in the ATU FID register (Global 1 offset 0x01).

The procedure to collect ATU Statistics is as follows:

1. Set the kind of statistics to collect in the ATU Stats register (Global 2 offset 0x0E).
2. Define the ATU FID to count (if needed in Global 1 offset 0x01) and then issue an ATU GetNext ATUOp and wait for it to finish (Global 1 offset 0x0B).
3. Read the statistics results from each of the 4 bins (Global 2 offset 0x0E).

9.2 VLAN Translation Unit Operations

The VLAN Translation Unit (VTU) in the device supports user commands to access and modify the contents of the VLAN membership database.

All VTU operations have the same user interface and protocol. Global registers are used and are shown in Table 46. The protocol for an VTU operation is as follows:

- Ensure the VTU is available by checking the VTUBusy bit in the VTU Operation register. The VTU can only perform one user command at a time.
- Load the VTU Data and VTU VID registers if required by the desired operation.
- Start the VTU operation by defining the required FID, and VTUOp and setting the VTUBusy bit to a one in the VTU Operation register – this can be done with a single write operation.
- Wait for the VTU operation to complete. This can be done by polling the VTUBusy bit in the VTU Operation register or by receiving an VTUDone interrupt (see Switch Global Control, global offset 0x04, and Global Status, offset 0x00).
- Read the required results if appropriate.

.

Table 46: VTU Operation Registers

Register	Offset	Before the Operation Starts	After the Operation Completes
VTU FID	0x02	Used to define which address database is to be associated with this VID and if this VID is a Policy VID	Used to indicate the returned VID's FID and Policy.
VTU SID	0x03	Used to define which 802.1s instance is to be associated with this VID	Used to indicate the returned VID's SID
VTU Operation	0x05	Used to define the required operation (including which database to associate with this VID) and start it.	Used to indicate the VTU's Busy status and violation status including source of the violation.
VTU VID	0x06	Used to further define the required operation and used as the VID that is to be operated on.	Returns the VID from the desired operation.
VTU Data (3 registers)	0x07 to 0x09	Used to define the required data that is to be associated with the required VID, including VTU Priority Override.	Returns the associated data from the desired operation.

9.2.1 Format of the VTU Database

Each VID entry in the VTU database contains:

- A 1-bit valid indicator (Valid)
- A 12-bit FID (Forwarding Information Database) number
- A 6-bit SID (802.1s Information Database) number
- 4-bits of VTU Priority Override data (VIDPri[2:0] & UseVIDPri)
- 2 bits of VTU Data per port

The format of a VTU entry is shown in Figure 53 and Table 47. The database is accessed 16-bits at a time via the Switch Global registers shown in Table 47 (not all the register bits are shown). For more information about these register see the VTU Operation register (Global 1 offset 0x05) and VTU Data registers for all ports (Global 1 offsets 0x02, 0x03 and 0x06 to 0x09).

Figure 53: Format of a VTU Entry

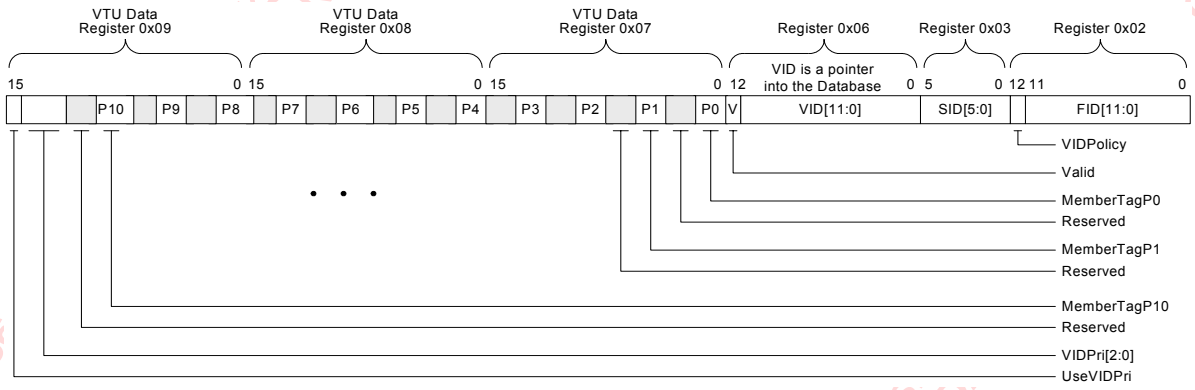


Table 47: VTU Entry Format

Field	Bits	Description
UseVIDPri	15 in Reg 0x09	VID Priority Override. This bit is used to indicate that frames assigned with this VID can have their priority overridden with the VIDPri bits below if either of the port's VTUPriOverride bits is set (see Port offset 0x08).
VIDPri	14:12 in Reg 0x09	VID Priority override value when enabled by the UseVIDPri bit above. Used for priority override on ingressing frames (see Section 5.4.6). Enabling a priority on a VID will override the frame's priority only if the port's VTUPriOverride bits are configured to do so (Port offset 0x0C).

Table 47: VTU Entry Format (Continued)

Field	Bits	Description
Reserved for 802.1s Port State	In Reg 0x07: Port 0 3:2 Port 1 7:6 Port 2 11:10 Port 3 15:14 In Reg 0x08: Port 4 3:2 Port 5 7:6 Port 6 11:10 Port 7 15:14 In Reg 0x09 Port 8 3:2 Port 9 7:6 Port 10 11:10	These bits are NOT part of the VTU and are NOT associated with the VID. They are used to access the STU database (802.1s per VLAN spanning tree – Section 5.2.3). On writes to the VTU these bits are don't care. On reads from the VTU these bits will not be updated and will contain the data from the last STU read (Section 9.2.6).
MemberTag	In Reg 0x07: Port 0 1:0 Port 1 5:4 Port 2 9:8 Port 3 13:12 In Reg 0x08: Port 4 1:0 Port 5 5:4 Port 6 9:8 Port 7 13:12 In Reg 0x09 Port 8 1:0 Port 9 5:4 Port 10 9:8	The lower two bits of each port's VTU data is called MemberTag. These bits are used to indicate which ports are members of the VLAN and if these VLANs frames should be tagged or untagged, or unmodified when exiting the port as follows 00 = Port is a member of this VLAN and frames are to egress unmodified 01 = Port is a member of this VLAN and frames are to egress Untagged 10 = Port is a member of this VLAN and frames are to egress Tagged 11 = Port is not a member of this VLAN
Valid	12 in Reg 0x6	Valid bit. This bit is used to indicate that the below VID and its associated data is valid in the VTU's database and should be used. After a hardware reset, all 4096 entries in the table are considered invalid (the Valid bit on each entry is cleared).
VID	11:0 in Reg 0x6	VLAN ID. These bits indicate the VID number that is associated with the MemberTag data, VTU Priority and its override, VTU Policy and the entry's Forwarding Information Database number (FID).
SID	5:0 in Reg 0x03	802.1s Information Database Number. If 802.1s per VLAN spanning tree is being used, these bits indicate the spanning tree instance number to use for all frames assigned with this VID (see Section 5.2.3). Multiple VID's can use the same SID. If per VLAN spanning tress are not used the SID must be written as zeros.
VIDPolicy	12 in Reg 0x02	VID Policy Entry. This bit is used to indicate that frames assigned with this VID can have Layer 2 Policy actions applied to it if either of the port's VTU Policy bits is set (see Section 5.1.3.1 and Port offset 0x0E).
FID	11: 0 in Reg 0x02	Forwarding Information DataBase Number. If separate address databases are used, these bits indicate the address database number to use for all frames assigned with this VID (see Section 4.4.8). All MAC DA look-ups and SA learning will refer to the address database number defined by the FID associated with the frame's VID. Multiple VID's can use the same FID. If separate address databases are not used the FID must be written as zeros.

9.2.2 Reading the VLAN Database

The contents of the VLAN database can be dumped or searched. The dump operation is called VTU Get Next since it returns the active contents of the VLAN database in ascending VID order. A search operation can also be done using the VTU Get Next operation.

The VTU Get Next operation starts with the VID contained in the VTU VID register and returns the next higher active VID in the VLAN database. Use a VID of all ones to get the first or lowest active VID. The returned VID and its data are accessible in the VTU Operation, VTU VID, VTU FID, VTU SID and the VTU Data registers. To get the next higher active VID, the VTU Get Next operation can be started again without setting the VID registers since it already contains the last address. A returned VID of all ones indicates that no higher active VID was found or that the VID value of 0xFFF was found. In either case, it indicates that the end of the database has been reached. If it were reached with a valid VID of 0xFFF the entry's Valid bit is returned set to one. A summary of how the VTU Get Next operation uses the VTU's registers is shown in Table 48.

Table 48: VTU Get Next Operation Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
VTU Operation	0x05	Used to define the required operation and start it.	Used to indicate the VTU's Busy status.
VTU VID	0x06	Used to define the starting VID to search. Use VID of all ones to find the first or lowest VID. Use the last address to find the next address (there is no need to write to this register in this case)	Returns the next higher active VID if found, or all ones are returned indicating the end of the table has been reached (all ones is a valid entry if the Valid bit = 1)
VTU Data (3 registers)	0x07 to 0x09	Ignored	Returns the VTU Data (lower 2 bits for each port) that is associated with the VID above. If the Valid bit = 0 the returned data is not a valid entry.
VTU FID	0x02	Ignored	Returns the VTU FID that is associated with the VID above. If the Valid bit = 0 the returned data is not a valid entry.
VTU SID	0x03	Ignored	Returns the VTU SID that is associated with the VID above. If the Valid bit = 0 the returned data is not a valid entry.

To search for a particular VID, start the VTU Get Next operation with a VID one less than the target VID. If the target VID is found, it is returned in the VTU VID register along with its associated data in the VTU Data register and VTU Operation register. If the target VID is not found active, the VID register contents do not equal the target VID.

9.2.3 Loading and Purging an Entry in the VLAN Database

Any VID can be loaded into or removed from the VLAN database by using the VTU Load/Purge operation. A VID is loaded into the database if the Valid bit in the VTU VID register (Global 1 offset 0x06) is a one. A value of zero in the Valid bit indicates that the VTU operation is a purge and that the defined VID and its data are to be removed from the database (if they exist).

The Load operation accesses the VLAN database using the VID contained in the VTU VID register. If the VID in the database is found valid, it is updated by the information found in the VTU Data registers and the VTU FID and SID registers.

Note

A load operation becomes a purge operation if the VTU Valid bit equals zero causing the entry's Valid bit to be cleared.

If the VID in the database is not valid, and if the VTU Valid bit equals one, then the VID, along with its data, will be loaded into the VLAN database.

A summary of how the Load operation uses the VTU's registers is shown in Table 49.

Table 49: VTU Load/Purge Operation Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
VTU Operation	0x05	Used to define the operation and start it.	Used to indicate the VTU's Busy status.
VTU VID	0x06	Used to define the VID to load or purge and to define if the operation is a load or a purge (Valid = 1 means load).	No change.
VTU Data (3 registers)	0x07 to 0x09	Used to define the associated Data (lower 2-bits for each port) that will be loaded with the VID above.	No change.
VTU FID	0x02	Used to define the associated FID that will be loaded with the VID above.	No change.
VTU SID	0x03	Use to define the associated SID that will be loaded with the VID above.	No change.

9.2.4 Flushing Entries

All VID's in the VLAN database can be purged by a singe Flush All Entries VTU Operation. The VTU VID, VTU FID, VTU SID, and VTU Data registers are not used by the Flush command.

Note

When the VTU is flushed the STU is also flushed (Section 9.2.8)

9.2.5 Servicing VTU Violations

The VTU captures VID Member Violation and VID Miss Violation data. A VID Membership Violation occurs when an 802.1Q enabled port receives a frame whose VID is contained in the VLAN database (VTU) but where the source port is not a member of that VLAN. A VID Miss Violation occurs when an 802.1Q enabled port receives a frame whose VID is not contained in the VLAN database (VTU).

Captured VTU Violations and their associated interrupts are cleared by the Get/Clear Violation Data VTU Operation. This VTU Operation returns the source port number that caused the violation in the SPID field of the VTU Operation register (Global 1 offset 0x05) and returns the VID that caused the violation in the VID field of the VTU VID register (Global 1 offset 0x06).

A summary of how the Get/Clear Violation Data operation uses the VTU's registers is shown in Table 50.

Table 50: VTU Get/Clear Violation Data Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
VTU Operation	0x05	Used to define the desired operation and start it.	Used to indicate the VTU's Busy status, the type of violation and the source port of the violation.
VTU VID	0x06	Ignored	Used to indicate the VID that was involved in the violation
VTU Data (3 registers)	0x07 to 0x09	Ignored	No change
VTU FID	0x02	Ignored	No change.
VTU SID	0x03	Ignored	No change.

9.2.6 Format of the STU Database

Each SID entry in the STU database contains:

- A 1-bit valid indicator (Valid)
- A 6-bit SID (802.1s Information Database) number
- 2 bits of STU Data per port

The format of an STU entry is shown in Figure 54 and Table 51. The database is accessed 16-bits at a time via the Switch Global registers shown in Figure 54 (not all the register bits are shown). For more information about these register see the VTU Operation register (Global 1 offset 0x05) and VTU Data registers for all ports (Global 1 offsets 0x03 and 0x06 to 0x09).

Note

The STU is accessed using many of the VTU registers.

Figure 54: Format of an STU Entry

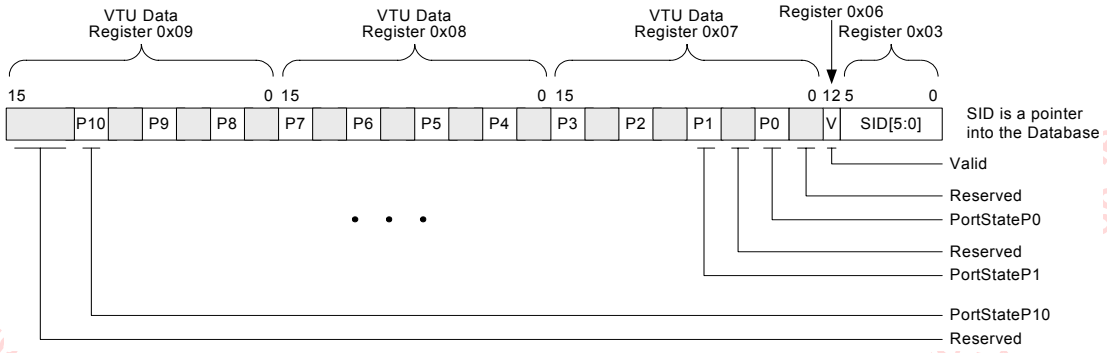


Table 51: VTU Entry Format

Field	Bits	Description
Reserved for VTU Priority Override	In Reg 0x09: 15:12	These bits are NOT part of the STU and are NOT associated with the SID. They are used to access the VTU database (802.1Q VLANs – Section 5.2.2). On writes to the STU these bits are don't care. On reads from the STU these bits will not be updated and will contain the data from the last VTU read (Section 9.2.2).

Table 51: VTU Entry Format (Continued)

Field	Bits	Description
802.1s Port State	In Reg 0x07: Port 0 3:2 Port 1 7:6 Port 2 11:10 Port 3 15:14 In Reg 0x08: Port 4 3:2 Port 5 7:6 Port 6 11:10 Port 7 15:14 In Reg 0x09 Port 8 3:2 Port 9 7:6 Port 10 11:10	The upper two bits of each port's VTU data register is called 802.1s PortState. These bits are used to support 802.1s per VLAN spanning tree as follows: 00 = 802.1s Disabled. Use non-VLAN Port States (i.e., the port's default Port State - Port offset 0x04) for this port for frames with a VID that is associated to this SID 01 = Blocking/Listening Port State for this port for frames with a VID that is associated to this SID 10 = Learning Port State for this port for frames with a VID that is associated to this SID 11 = Forwarding Port State for this port for frames with a VID that is associated to this SID These 802.1s PortState bits take precedence over the port's Port State bits (Port offset 0x04) unless the port's Port State is Disabled (which prevents all frames from flowing).
Reserved for MemberTag	In Reg 0x07: Port 0 1:0 Port 1 5:4 Port 2 9:8 Port 3 13:12 In Reg 0x08: Port 4 1:0 Port 5 5:4 Port 6 9:8 Port 7 13:12 In Reg 0x09 Port 8 1:0 Port 9 5:4 Port 10 9:8	These bits are NOT part of the STU and are NOT associated with the SID. They are used to access the VTU database (802.1Q VLANs – Section 5.2.2). On writes to the STU these bits are don't care. On reads from the STU these bits will not be updated and will contain the data from the last VTU read (Section 9.2.2).
Valid	12 in Reg 0x06	Valid bit. This bit is used to indicate that the below SID and its associated data is valid in the STU's database and should be used. After a hardware reset, all 64 entries in the table are considered invalid (the Valid bit on each entry is cleared).
SID	5:0 in Reg 0x03	802.1S Instance Database number. These bits indicate the SID number that is associated with the 802.1s Port State bits above.

9.2.7 Reading the SID Database

The contents of the STU database can be dumped or searched. The dump operation is called STU Get Next since it returns the active contents of the STU database in ascending SID order. A search operation can also be carried out using the STU Get Next operation.

The STU Get Next operation starts with the SID contained in the VTU SID register and returns the next higher active SID in the STU database. Use a SID of all ones to get the first or lowest active SID. The returned SID and its data are accessible in the VTU Operation, VTU SID and the VTU Data registers. To get the next higher active SID, the STU Get Next operation can be started again without setting the SID registers since it already contains the last address. A returned SID of all ones indicates that no higher active SID was found or that the SID value of 0x3F was found. In either case, it indicates that the end of the database has been reached. If it were reached with a valid SID of 0x3F the entry's Valid bit is returned set to one. A summary of how the STU Get Next operation uses the VTU's registers is shown in Table 52.

Note

The STU is accessed using many of the VTU Registers.

Table 52: STU Get Next Operation Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
VTU Operation	0x05	Used to define the required operation and start it.	Used to indicate the STU's Busy status.
VTU VID	0x06	Ignored	Returns the Valid bit = 1 if a valid SID was found or 0 = if the end of the table was reached and the last entry was unused.
VTU Data (3 registers)	0x07 to 0x09	Ignored	Returns the STU Data (upper 2 bits for each port) that is associated with the SID below. If the Valid bit = 0 the returned data is not a valid entry.
VTU FID	0x02	Ignored	Ignored
VTU SID	0x03	Used to define the starting SID to search. Use SID of all ones to find the first or lowest SID. Use the last address to find the next address (there is no need to write to this register in this case)	Returns the next higher active SID if found, or all ones are returned indicating the end of the table has been reached (all ones is a valid entry if the Valid bit = 1)

To search for a particular SID, start the STU Get Next operation with a SID one less than the target SID. If the target SID is found, it is returned in the VTU SID register along with its associated data in the VTU Data register and VTU Operation register. If the target SID is not found active, the SID register contents do not equal the target SID.

9.2.8 Loading and Purging an Entry in the STU Database

Any SID can be loaded into or removed from the STU database by using the STU Load/Purge operation. A SID is loaded into the database if the Valid bit in the VTU VID register (Global 1 offset 0x06) is a one. A value of zero in the Valid bit indicates that the STU operation is a purge and that the defined SID and its data are to be removed from the database (if they exist).

The Load operation accesses the STU database using the SID contained in the VTU SID register. If the SID in the database is found valid, it is updated by the information found in the VTU Data registers.

Note

A load operation becomes a purge operation if the STU Valid bit equals zero causing the entry's Valid bit to be cleared.

If the SID in the database is not valid, and if the register's Valid bit equals one (Global 1, offset 0x06), then the SID, along with its data, will be loaded into the STU database. A summary of how the Load operation uses the VTU's registers is shown in Table 53.

Note

The STU is accessed using many of the VTU registers.

Table 53: STU Load/Purge Operation Register Usage

Register	Offset	Before the Operation Starts	After the Operation Completes
VTU Operation	0x05	Used to define the required operation and start it.	Used to indicate the STU's Busy status.
VTU VID	0x06	Used to define if the operation is a load or a purge (Valid = 1 means load).	No change.
VTU Data (3 registers)	0x07 to 0x09	Used to define the associated Data (upper 2-bits for each port) that will be loaded with the SID below.	No change.
VTU FID	0x02	Ignored	No change.
VTU SID	0x03	used to define the SID to load or purge.	No change.

9.2.9 Flushing Entries

All SID's in the STU database can be purged by a singe Flush All Entries VTU Operation. The VTU VID, VTU FID, VTU SID and VTU Data registers are not used by the Flush command.

Note

When the STU is flushed the VTU is also flushed (Section 9.2.4)

Section 10. Remote Management

Remote Management is a method of accessing Switch registers using Ethernet frames. It is intended to be an alternate way of accessing registers, in addition to the original way they are already accessed (i.e., using System Management Interface (SMI), made up of the MDC and MDIO signals or the EEPROM).

The benefits of Remote Management are:

- Faster CPU access to large databases of information – specifically the address database (i.e., the ATU – Section 9.1) and the port statistics (i.e., the MIBs – Section 4.3.8 and Section 4.3.9).
- Uses the CPU's frame interface (e.g., MII or GMII) to access switch registers saving the need for an alternate interface and its pins on the CPU device.
- Remote access to all switch registers without the need of a local CPU.
- Can control which physical port accepts and processes Remote Management requests (for security).
- Request/Response type protocol. The CPU asks for data only when it wants data. Other methods can be intrusive by cramming information into the CPU at potentially inopportune times.

The Remote Management Unit (RMU) is initially disabled at reset. It needs to be enabled by a CPU using the SMI interface or by an EEPROM attached to the device. The RMU is enabled by setting the RMEnable bit to a one (Global 1 offset 0x1C) and by selecting either Port 9 or Port 10 as the port to accept and process RMU frames (using the P10RM bit in Global 1 offset 0x1C). Additionally the RMU can be configured to ignore all RMU frames it receives unless the frame's Destination Address (DA) is loaded as static in the address database (Section 9.1.1). This option is enabled by setting DA Check to a one (Global 1 offset 0x1C).

The RMU can only be enabled on one physical port at a time for security reasons. It needs to be enabled on the port that is directly or indirectly connected to the switch management CPU. The port must also be in either DSA Tag mode or Ether type DSA Tag mode (Section 7).

Note

The RMU sits outside the switch core. This ensures the RMU can accept and process frames even if the switch core cannot. It snoops all frames coming in the enabled RMU port and it will transmit RMU response frames ahead of any other frames in the port's egress queue.

10.1 Request for Frame Format - Layer 2 and DSA Portion

Remote Management is a Request/Response protocol so the CPU needs to send a Request frame to the switch. Once outside the CPU, the frame needs to get to the desired physical switch device chip.

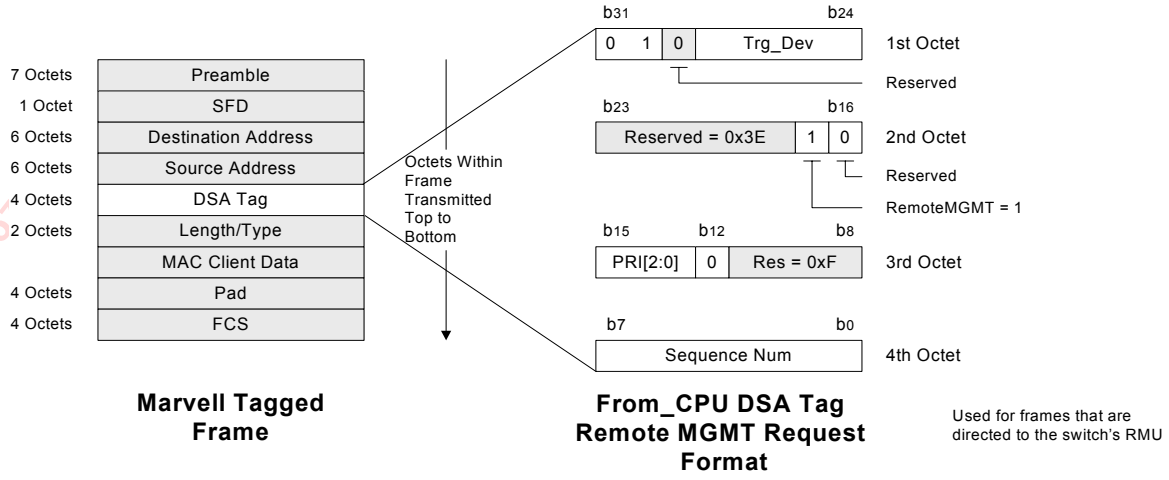
The layer 2 portion of the Remote Management frame contains the normal IEEE 802.3 fields of DA, SA, etc. The DA of these frames is defined to be the Marvell® multicast address of 01:50:43:00:00:00[1] or the unicast address of the switch[2]. The SA is the MAC address of the source device. The Distributed Switch Architecture (DSA) portion of the request frame is defined in Figure 55 and the Ether Type (Length/Type) portion of the frame is user definable[3]. The DSA portion may optionally be Ether Typed DSA Tagged[4] (Section 7.9).

A From_CPU type DSA Tag frame format is used as these frames contain a Trg_Dev field that all DSA enabled devices support that is used to get these frames to the desired switch device if more than one of these devices are interconnected in a box using DSA enabled links. Bit 17 being a one, along with all the other reserved bits being the correct value, define the frame to be for Remote Management.

Note

All Reserved bits must be the values defined in Figure 55. This means that bits 29, 18, 16 & 12 must be zero and bits 23:19, 17 & 11:8 must be one in the frame.

Figure 55: Remote Management DSA Tag Request Format



Older devices that do not understand this frame format will simply map these frames to the Trg_Dev (and hopefully to a device that does understand the Remote Management function). If Trg_Dev is this device, it will process the frame if all the other DSA Tag bits are the correct value, and then the frame will be discarded.

1. The DA is defined to be a Marvell multicast for the remote control protocol discovery cases where the frame may need to pass through some other L2 switch(es) before the frame gets to the target device. Inside the Marvell switches the DA can be ignored (for non-remote used) or validated (i.e., a required value for remote use).
2. A unicast address is desirable for remote controlling of more than 32 devices. To discover the SA address of the switch a GetID request is used with the multicast DA and all attached switches that support Remote Management will respond with their individual SA's. In either case the DA validation of Remote Management frames should be enabled for remote use. DA validation requires that the DA of Remote Management frames (unicast or multicast) are present in the ATU as Static entries or the frame will not be considered a Remote Management frame by this device.
3. The Length/Type field is NOT validated by the hardware.
4. The support of Ether Type DSA may be needed in remote control cases where a non-Ether Type DSA Tag could confuse other L2 switch(es) the frame may need to pass through.

Using the From_CPU DSA Tag gets the Remote Management Request frames to the correct device. This works the same way for DSA and EtherTyped DSA (Section 7).

The usage of the DSA Tag's PRI and Sequence Num is discussed in Section 10.2.

10.1.1 RMU and Ether type DSA

Supporting the Ether typed DSA frame format (Section 7.9) can be used for remote control of a switch using a single link, something a ISP would do to communicate with a remote switch on the outside of someone's house. This application requires the same 'pipe' to be used for both customer data and these Remote Management frames. Security can be assured by allowing Ether typed DSA frames from the provider port only and never from the customer's port(s). For additional security the device supports processing of Remote Management frames only if they ingress a defined physical port (Port 9 or Port 10). This requires that the 'defined' port for Remote Management be either directly or indirectly connected to the management CPU's port. An indirect connection is one where a Remote Management frame hops through one or more other DSA enabled devices before it gets to this device. And the port it enters on this device is the port that is enabled for Remote Management frame processing.

10.1.2 RMU and Marvell® Header

Remote Management is supported on a port where DSA or Ether type DSA frame mode is enabled (Section 7). It is also supported on a local CPU's port where DSA or Ether type DSA frame mode is enabled together with the Marvell Header Mode being enabled (Section 8.7). The Marvell Header mode is used to accelerate routing by aligning the layer 3 portion of the Ethernet frame onto a 32-bit boundary. This is done by inserting 2 bytes before the frame's DA.

When Remote Management is used on a port where the Marvell Header mode is enable, all ingressing frames, even the Remote Management frames, must contain the extra 2 byte Header before the ingressing frame's DA. In this case it is recommended that the extra 2 bytes be zeros.

Refer to Section 10.2.1 for restrictions with the use of the Marvell Header mode with Remote Management.

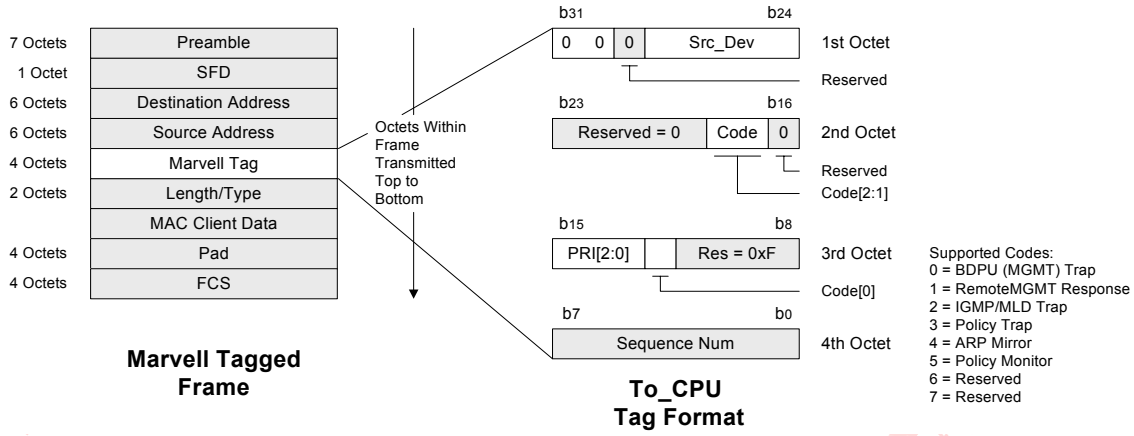
10.2 Response Frame Format - Layer 2 and DSA Portion

Remote Management is a Request/Response protocol so the CPU needs to get a Response frame back from the switch to return any requested data and to validate that the switch received the Request frame, e.g., it wasn't discarded somewhere along the way (clearly needed if the request was only register writes). All valid Remote Management requests are acknowledged with a response so the software knows the request was received and acted on.

The layer 2 portion of the Remote Management frame contains the normal IEEE 802.3 fields of DA, SA, etc. The DA of these frames is defined to be the SA from the Request frame. The DSA portion of the frame is defined in Figure 56 and the Ether Type (Length/Type) is a copy of the Length/Type field of the Request frame (Figure 55).

A To_CPU DSA Tag frame format is used as these frames are automatically mapped back to the CPU.

Figure 56: Remote Management DSA Tag Response Format



The priority of the response frame (PRI bits) are the PRI bits from the request frame. The request frames are assumed to come from a trusted source and are assumed to be required for proper management of the switch. This assumption is policed by ensuring only one port on the switch is allowed to process received Remote Management frames. It is further expected that these frames should be infrequent and important and thus they should use the highest priority.

The Sequence Num extracted from the Request frame (Figure 55) is used as the Sequence Num in the Response frame. It is a way for the CPU to match up Responses to Requests in case a Request or a Response gets dropped.

10.2.1 Restrictions of Remote Management

The RMU is outside of the switch core and does not pass response frames through the switch (i.e., through the queue controller). Instead, the frames are built in a separate SRAM and then MUX'ed and transmitted out a port like a Pause frame (i.e., in front of the normal data, momentarily stalling that port's Tx path). Assuming these frames would normally be mapped to the highest priority queue defines that these 'built' response frames are be transmitted ahead of all other frames currently in the port's output queue.

The separate SRAM approach limits the ability of receiving, processing and then transmitting more than one Remote Management frame at a time. If a 2nd Remote Management Request frames is sent to the device before the device has completely transmitted the 1st Response frame, the 2nd (and subsequent) Request frames will be dropped until the 1st Request has been completely transmitted.

In the device, a separate 512 byte SRAM is used that can be MUX'ed in or out Port 9 or Port 10 (register selectable – P10RM in Global 2 offset 0x1C). The 512 byte SRAM yields a maximum Response Data size of 484 bytes.

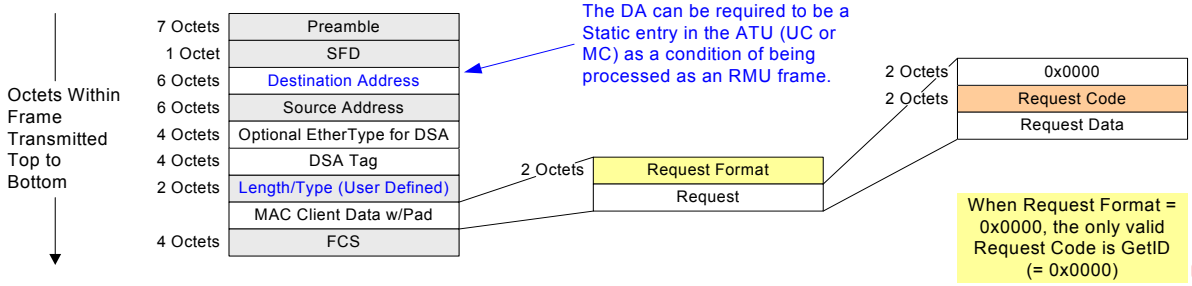
Since a separate SRAM is used in the device, the Marvell® Header will not be added to Remote Management response frames that egress a port where the Marvell Header is enabled. The Marvell Header is properly removed from Remote Management request frames that enter the port and the SRAM receives these modified frames for processing. So the Marvell Header must be present on ingressing request frames if the Header mode is enabled on the port. The Header just won't be added to egressing response frames on these ports (but it will be added on all other frames that egress this port).

Software will need to look for the From_CPU RemoteMGMT Response code two bytes ahead of where they normally would be if the Header was added to the egressing frame. Since CPU NIC's should not remove the Header when it is there, the full content of these frames will still make it into the CPU's memory. Some CPU's know how to parse the Header to determine which CPU memory receive queue the frame should be mapped into. In this case the 1st two bytes of the DA will be processed as the Header. Since the DA is known (it was the CPU's SA) the receive queue mapping will be known too so the CPU will know which receive queue the RemoteMGMT response frames will be found.

10.3 Request Frame Format - Layer 3

The layer 3 portion of the Remote Management request frame contains the specific register action requests. These requests are encapsulated behind a Request Format (Figure 57). The Request Format supports various device families that require very different command and/or response formats due to the internal nature of the specific device family.

Figure 57: Remote Management Generic Layer 3 Request Format



10.3.1 The Initial Request Frame - GetID

The initial request frame uses a Request Format = 0x0000 with a Request Code = 0x0000 and the rest of the frame's Request Data field being padded with 0x00 bytes. The intention of this initial request frame (called GetID) is to return the Request and Response Format number supported by the addressed device and the unicast address of the device. It can also be used by the requestor to determine the latency in the 'system' by timing the time it takes to get the response back. This can be used to help tune software timeouts for cases when frames are dropped.

The device supports the GetID request and it is the only Request Code allowed under the 0x0000 Request Format.

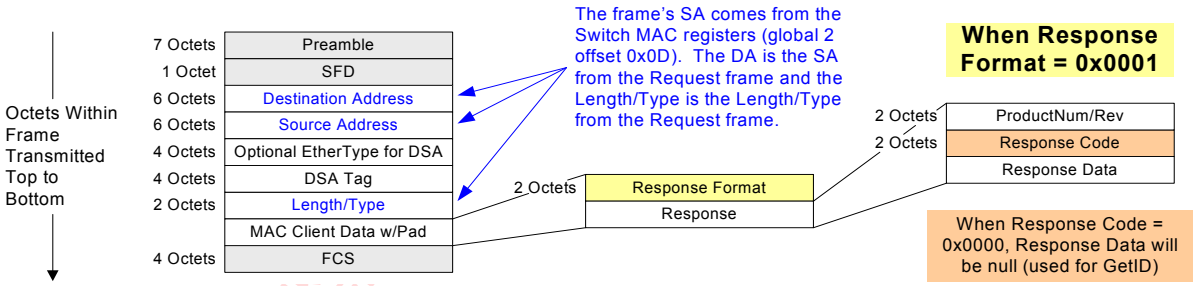
One of the fields returned in the response to a GetID (a GotID – Section 10.4.1) is the Request Format. This device uses a Request Format of 0x0001 for all other commands.

The Request Format is intended to be a Family code where devices that contain the same register interface structure will all be of the same Family. All Marvell® SOHO devices have a similar register interface structure and so they all use the same Request Format even though there are major differences between the SOHO devices. These product specific differences are indicated by the ProductNum/Rev field returned in all Response frames (included in the GotID response to the GetID request). The Response Format is described in Section 10.4.

10.4 Response Frame Format - Layer 3

The layer 3 portion of the Remote Management response frame contains the requested data. These responses are encapsulated behind a Response Format (Figure 58). The Response Format supports various device families that may need very different command and/or response formats due to the internal nature of the specific device family.

Figure 58: Remote Management Generic Layer 3 Response Format



10.4.1 The Initial Response Frame - GotID

The response to the initial request frame of GetID (Section 10.3.1) is a GotID response. A GotID response contains the device's Response Format number followed by the Product Number and Revision of the device, followed by the Response Code and Response Data.

This device supports a Response Format of 0x0001. In this case the ProductNum/Rev is the ProductNum/Rev from the Switch Identifier register (Port offset 0x03). The Response Code for the GotID is 0x0000 and the Response Data is null.

10.4.2 Error Handling

The layer 3 portions of the Request frame's Request Code are fully decoded and all frames (that make it into the Remote Management processing unit) are acknowledged with a Response frame. If the CPU sends in a Request Code that is undefined, the hardware will respond with an Error Response frame (Section 10.5.6). This way the software will be able to determine that it got back data it did not expect.

10.5 Supported Requests and Responses

This device uses a Request and Response Format = 0x0001[1].

Non-destructive Remote Management requests can be done at the same time as register operations received on the device's SMI interface pins (MDC and MDIO). Likewise, if the SMI interface limits its actions to non-destructive actions then any Remote Management request can be done at the same time. A destructive action is any request that changes register data. Register writes fall into this category, but so do flush commands (like on the ATU, MIBs, etc.) as well as any register read that causes a bit to be cleared due to the read. Destructive Remote Management commands are noted.

The Request Codes supported in the device are:

- 0x0000 – GetID, Section 10.5.1
- 0x1000 – Dump ATU, Section 10.5.2
- 0x1020 – Dump MIBs, Section 10.5.3 (dump only) & Section 10.5.4 (dump & clear)
- 0x2000 – Read/Write Register, Section 10.5.5

10.5.1 GetID (non-destructive)

The purpose of this request is to get the Response Format and Product/Num Rev of the selected device.

Request Format = 0x0000 NOTE: this is 0x0000 for GetID only

Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x0000 (GetID)

Request Data = 0x0000

Response Format = 0x0001 (SOHO)

ProductNum/Rev = [Port 0, offset 0x03]

Response Code = 0x0000 (GotID)

Response Data = null

1. The GetID Request uses a Request Format = 0x0000 but its GotID Response uses the Response Format of the device (i.e., = 0x0001).

10.5.2 Dump ATU (non-destructive)

This request dumps up to 48 valid MAC entries found in the ATU. To start at the beginning of the ATU's memory use a Continue Code = 0x0000. If the ATU is empty the EntryState in the 1st Entry Data field will be 0x0 (bits 7:5 of the 1st octet). If less than 48 valid MAC entries are found, the 1st Entry with an EntryState = 0x0 is the end of the list. If more than 48 valid MAC entries exist in the ATU all 48 entries will be returned with a non-zero EntryState and a (two octet) Continue Code will appear after the 48th Entry Data. The next valid MAC entries can be retrieved by doing a Dump ATU using the Continue Code returned from the previous Dump ATU.

The ATU is dumped in linear ATU memory address order (i.e., they will not be in the GetNext's ascending network byte order) and only valid entries are dumped.

Request Format = 0x0001 (SOHO)

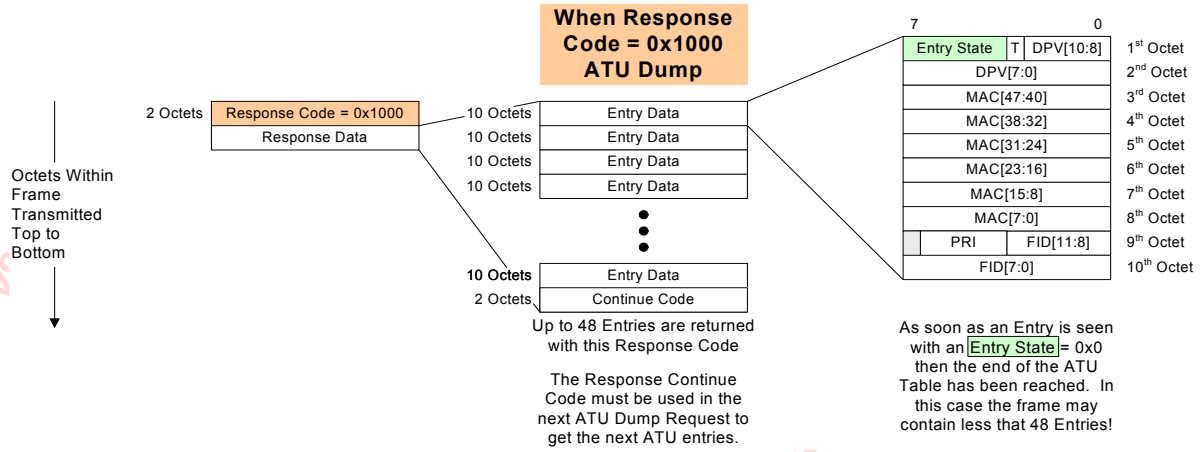
Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x1000 (ATU Dump)

Request Data = [Continue Code] size of 2 Octets (0x0000 = start at beginning of ATU table)

Response = see Figure 59.

Figure 59: Remote Management ATU Dump Response Format



Note

This ATU dump can be occurring in parallel to an ATU GetNext operation received from the SMI pins. This is not recommended to be done for more than the occasional[1] GetNext however, as the GetNext will slow down the ATU Dump's response time.

1. An occasional GetNext can be useful as a 'Search' function for a specific MAC address in the ATU. This approach will be faster than dumping the entire ATU contents with multiple frames (and the Search using the SMI pins can be done in parallel with an ATU Dump using a Remote Management frame).

10.5.3 Dump MIBs (non-destructive)

This request dumps all the current MIB counters for a single physical port on the selected device. The physical port number to dump is passed in the Request Data field defined below.

Request Format = 0x0001 (SOHO)

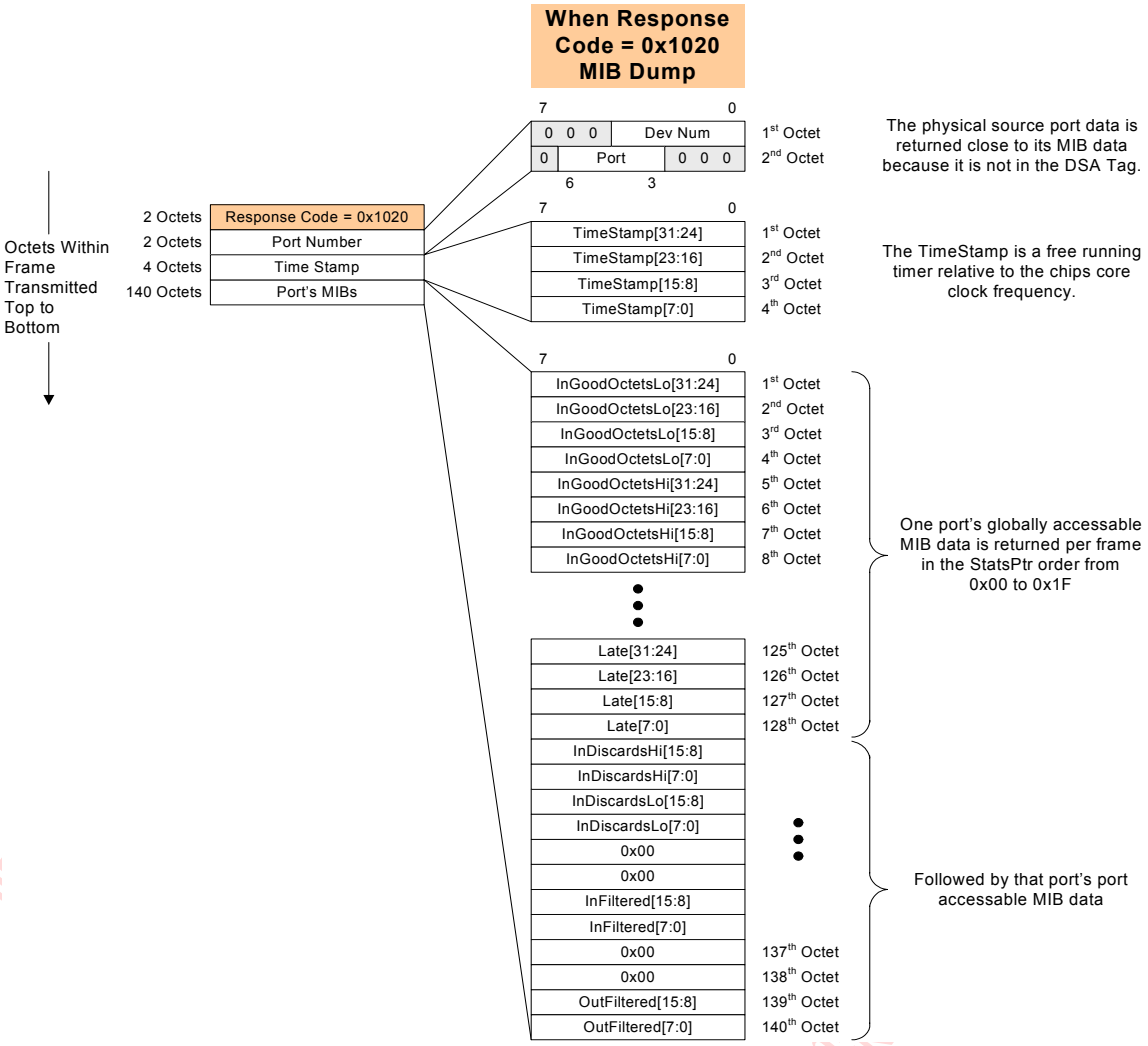
Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x1020 (MIB Dump)

Request Data = 0x000p (p = physical port number) In this device 0x0 => p <= 0xA

Response = see Figure 60.

Figure 60: Remote Management MIB Dump Response Format



The Port that is dumped is returned (along with the Dev Num – DeviceNumber from Global 1 offset 0x01C) so software knows what port this data is associated with without needing to look at the Sequence Num of the frame.

The TimeStamp is inserted into the frame at the instant the MIB values are just starting to be transferred into the frame. The delta time between two TimeStamp values can be used for rate calculations on the other MIB data.

In this device the TimeStamp increments at a rate of once every 512ns. At this rate the TimeStamp will overflow in about 2,200 seconds or over 36 minutes.

Note

The MIBs can be read with this Remote Management request in parallel with the SMI interface capturing and reading the MIBs.

10.5.4 Dump MIBs and Clear (destructive)

This request dumps all the current MIB counters for a single physical port on the selected device and then clears them. This way each Dump MIB & Clear will return the MIB count deltas from the last Dump MIB & Clear request for the same port. The physical port number to dump is passed in the Request Data field defined below.

Request Format = 0x0001 (SOHO)

Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x1020 (MIB Dump)

Request Data = 0x800p (p = physical port number – bit 15 = 1 indicates Clear after Read)

In this device 0x0 => p <= 0xA

Response = see Figure 60.

Note

This command is destructive due to the 'Clear' function and should not be done in parallel with the SMI interface accessing the MIBs. Choose one interface or the other

10.5.5 Read/Write Register (may be destructive)

This request reads or writes a single or multiple switch core or PHY registers.

Note

In this device the PHY registers (and any external registers connected to the device's MDC_PHY/ MDIO_PHY pins) are accessible only if the PPU is enabled. The switch core registers are always accessible (i.e., SMI Devices 0x10 to 0x1C in this device).

Caution

If this command is used to issue a SWReset a Response frame will NOT be generated as the MACs and data path are being reset. Other bit settings can be just as problematic, such as forcing the Link down or setting the port's PortState to Disabled on the port being used to receive and transmit Remote Management frames.

Request Format = 0x0001 (SOHO)

Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x2000 (Register read or write)

Followed by n sets of 4 Octet Register Commands where 0 < n < 121.

Three types of Register Commands are supported:

1. Direct Register read or write
   This command is used to read or write 16 bits to any device register as follows:

   1st Octet [7:4] = 0x0

   1st Octet [3:2] = Op Code where 10=Read and 01=Write

1st Octet [1:0] + 2nd Octet [7:5] = SMI Device Address

2nd Octet [4:0] = SMI Register Offset

3rd Octet = Data [15:8] (must = 0x00 for Reads)

4th Octet = Data [7:0] (must = 0x00 for Reads)

2. Wait on a Bit Command
   This command is used to delay processing the next Register Command in the frame until the bit selected is at the desired value. This is typically used to wait unit a register's Busy bit clears (like in the ATU or MIBs, etc.). The format of the Wait on a Bit Command is as follows:

   1st Octet [7:4] = 0x1

   1st Octet [3:2] = Op Code where 00=Wait until bit is a 0 & 11=Wait until bit is a 1

   1st Octet [1:0] + 2nd Octet [7:5] = SMI Device Address

   2nd Octet [4:0] = SMI Register Offset

   3rd Octet [7:4] = 0x0

   3rd Octet [3:0] = Bit in register to wait on (0x0 = bit 0, 0x1 = bit 1, …, 0xF = bit 15)

   4th Octet = 0x00

3. End of List Command
   This command is used to indicate no more commands occur in the frame. If a frame has the full 121 commands then the 121st must be an End of List. The format of the Wait on a Bit Command is as follows:

   1st Octet = 0xFF

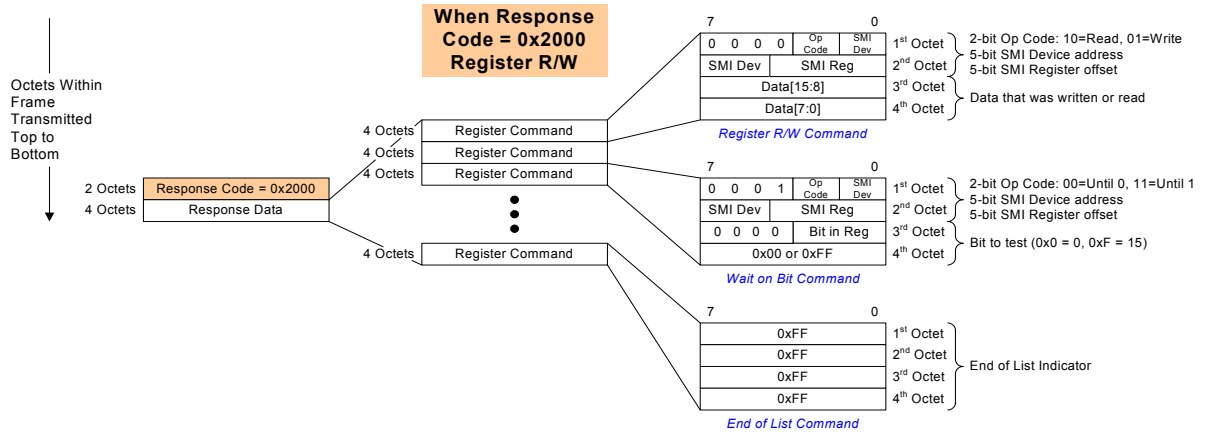
   2nd Octet = 0xFF

   3rd Octet = 0xFF

   4th Octet = 0xFF

Response = see Figure 61.

Figure 61: Remote Management Register Read/Write Response Format



The Response Frame's Response Data is the Request Frames' Request Data with the Data [15:0] portions of Register Read commands being the data that was read from the selected register (i.e., the 0x0000 data that was in the Request frame is replaced with the data read from the register).

Note

If a read or write register command is directed at a PHY register when the PPU is disabled, the operation will not occur and the read data returned will be 0xFFFF.

Any illegal register format is considered an End of List Command.

If a Wait on Bit command waits for a bit to be in the requested state for more than 1 second, the Wait on Bit command will terminate, return a value of 0xFF in the 4th Octet, and consider this instruction an End of List Indicator (i.e., any commands that follow the failed Wait on Bit command will not be processed).

Reading 'clear on read' registers is considered destructive and should not be done in parallel with destructive SMI operations.

Writing registers is also considered destructive. Only one interface (the Remote Management interface or the SMI interface) should be used to perform destructive operations (i.e., actions that cause changes inside the device).

10.5.6 Error Response Frame (non-destructive)

Any Request frame that contains an unsupported Request Format or Request Code is responded to with the Error Response frame (as ALL requests return a response).

Request Format = 0x[unsupported] (i.e., Request Format was not 0x0000 or 0x0001 for a SOHO device)

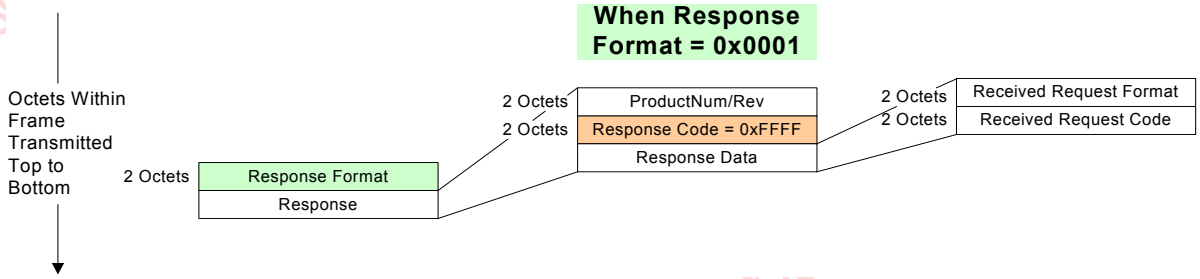
Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x[undefined] (i.e., the Request Code is an invalid one)

Request Data = 0x[don't care]

Response = See Figure 62.

Figure 62: Remote Management Error Response Format



Note

If software receives an Error Response with a Received Request Code that is valid for the device, then the Request frame's Response Format must have been in error or visa versa. Both are returned.

Illegal PHY accesses (a register read or write to a PHY register when the PPU is disabled) will not generate an Error Response frame. Instead, the write data will not be written and the read data will be returned with 0xFFFF.

Illegal Register Commands will not generate an Error Response frame. They will be interpreted as an End of List command.

An Error Response frame will not be generated if more than 121 Register Commands occur in a Register R/W Request frame. Only the first 121 Register Commands will be processed and then an automatic End of List command will be assumed.

Section 11. Physical Interface (PHY) Functional Description (P0 to P7)

The devices contain IEEE 802.3 100BASE-TX and 10BASE-T compliant media-dependent interfaces on Port 0 to Port 7 for support of Ethernet over unshielded twisted pair (UTP) copper cable. DSP-based advanced mixed signal processing technology supports attachment of up 150 meters of CAT 5 cable to each of these interfaces. An optional, per port, automatic MDI/MDIX crossover detection function gives true “plug and play” capability without the need for confusing crossover cables or crossover ports.

Figure 63: Device Transmit Block Diagram

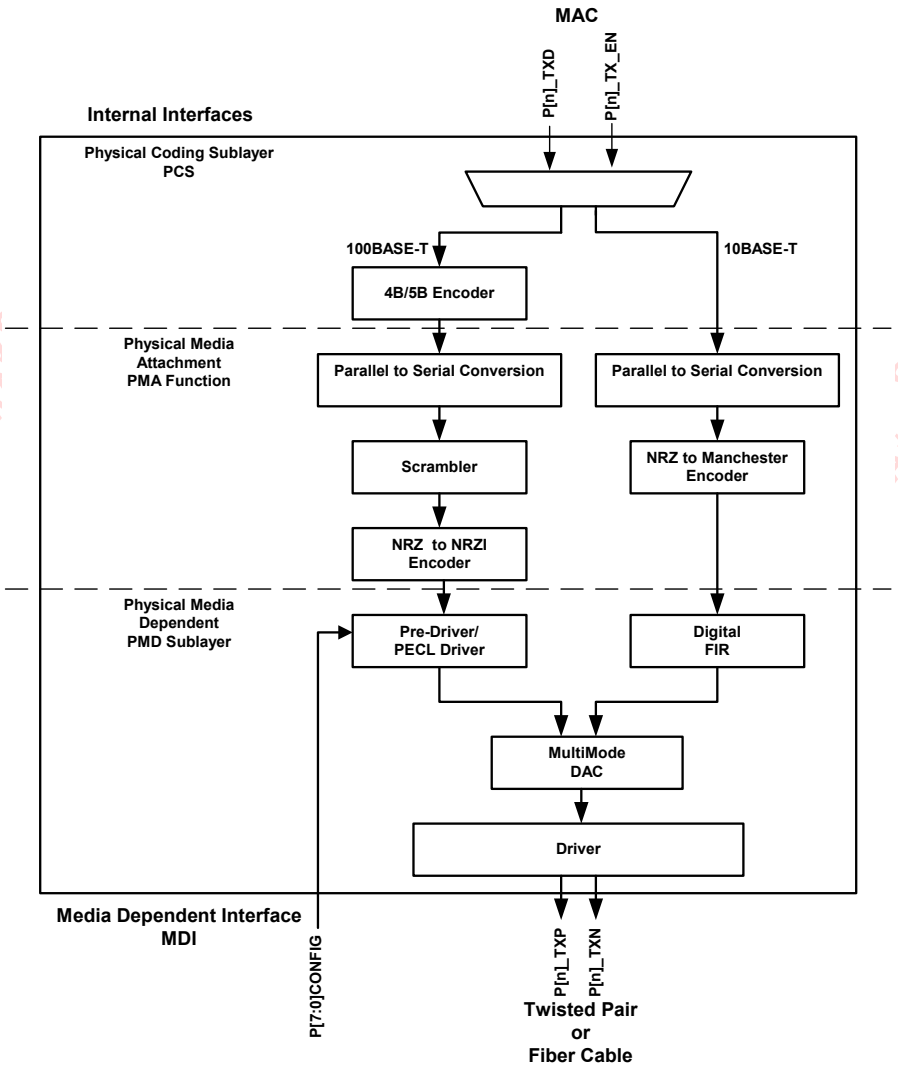
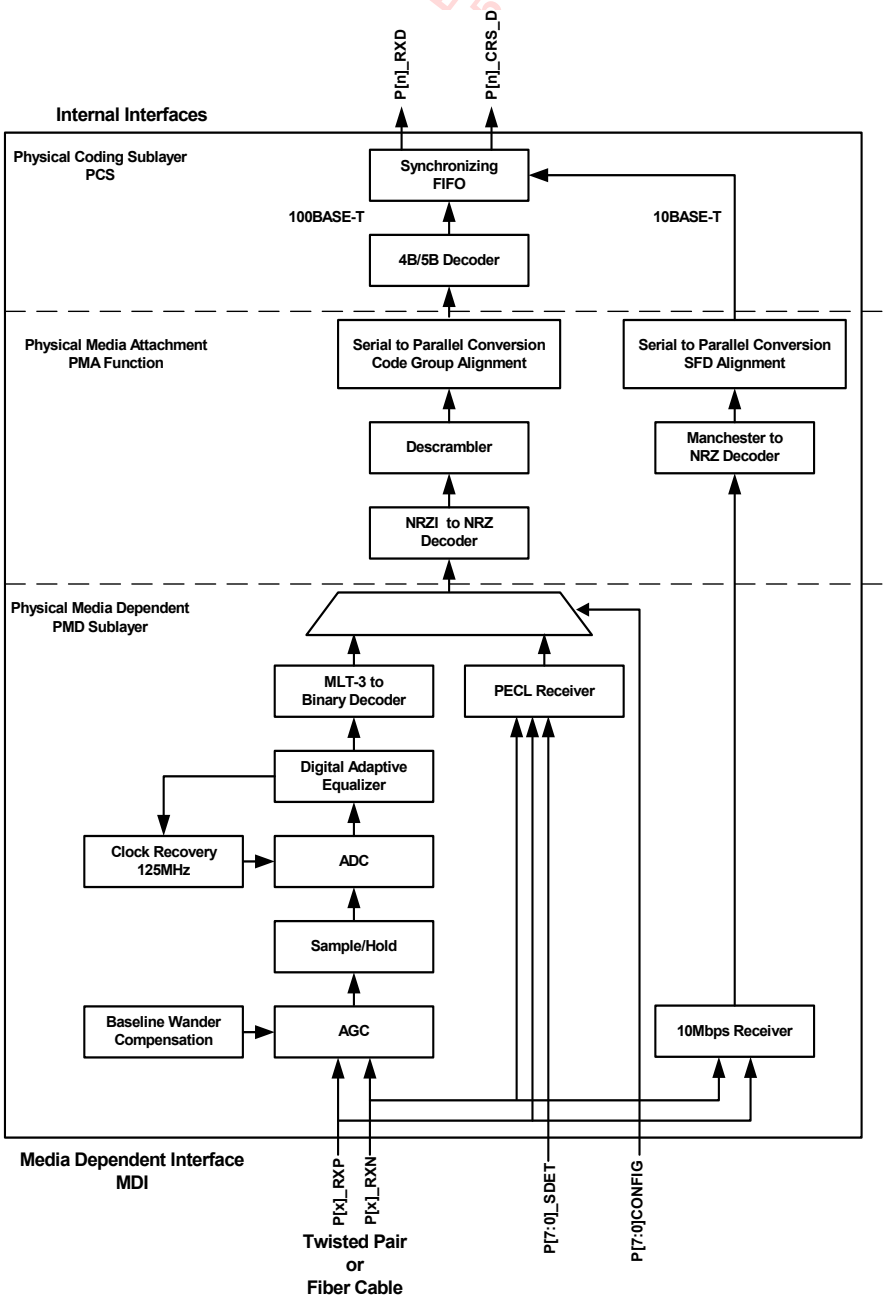


Figure 64: Device Receive Block Diagram



11.1 Transmit PCS and PMA

11.1.1 100BASE-TX Transmitter

The 100BASE-TX transmitter consists of several functional blocks that convert synchronous 4-bit nibble data to a scrambled MLT-3 125 Mbps serial data stream.

11.1.2 4B/5B Encoding

For 100BASE-TX mode, the 4-bit nibble is converted to a 5-bit symbol with /J/K/ start-of-stream delimiters and /T/R/ end-of-stream delimiters inserted as needed. The 5-bit symbol is then serialized and scrambled.

11.1.3 Scrambler

In 100BASE-TX mode, the transmit data stream is scrambled in order to reduce radiated emissions on the twisted pair cable. The data is scrambled by exclusive ORing the NRZ signal with the output of an 11-bit wide linear feedback shift register (LFSR), which produces a 2047-bit repeating pseudo-random sequence. The scrambler reduces peak emissions by randomly spreading the signal energy over the transmit frequency range and eliminating peaks at certain frequencies.

Note

The enabling and disabling of the scrambler and the far end fault generator are controlled in the same way as for the descrambler detection and far end fault detection on the receive side.

11.1.4 NRZ to NRZI Conversion

The data stream is converted from NRZ to NRZI.

11.1.5 Pre-Driver and Transmit Clock

The devices use an all-digital clock generator circuit to create the various receive and transmit clocks necessary for 100BASE-TX, 100BASE-FX, and 10BASE-T modes of operation.

For 100BASE-TX mode, the transmit data is converted to MLT-3-coded symbols. The digital time base generator (TBG) produces the locked 125 MHz transmit clock.

For 100BASE-FX mode, NRZI data is presented directly to the multimode DAC.

For 10BASE-T mode, the transmit data is converted to Manchester encoding. The digital time base generator (TBG) produces the 10 MHz transmit reference clock as well as the over-sampling clock for 10BASE-T waveshaping.

11.1.6 Multimode Transmit DAC

The multimode transmit digital to analog converter (DAC) transmits MLT-3 coded symbols in 100BASE-TX mode, NRZI symbols in 100BASE-FX mode, and Manchester-coded symbols in 10BASE-T mode. The transmit DAC utilizes a direct drive current driver which is well balanced to produce very low common mode transmit noise.

In 100BASE-TX mode, the multimode transmit DAC performs slew control to minimize high frequency EMI.

In 100BASE-FX mode, the pseudo ECL level is generated through external resistive terminations.

In 10BASE-T mode, the multimode transmit DAC generates the needed pre-equalization waveform. This pre-equalization is achieved by using a digital FIR filter.

11.2 Receive PCS and PMA

11.2.1 10-BASE-T/100BASE-TX Receiver

The differential RXP and RXN pins are shared by the 100BASE-TX, 100BASE-FX and Port 1 and 10BASE-T receivers.

The 100BASE-TX receiver consists of several functional blocks that convert the scrambled MLT-3 125 Mbps serial data stream to the synchronous 4-bit nibble data presented to the MAC interfaces.

11.2.2 AGC and Baseline Wander

In 100BASE-TX mode, after input to the AGC block the signal is compensated for baseline wander with a digitally controlled Digital to Analog converter (DAC). It automatically removes the DC offset from the input before it reaches the ADC input.

11.2.3 ADC and Digital Adaptive Equalizer

In 100BASE-T mode, an analog to digital converter (ADC) samples and quantizes the input analog signal and sends the result into the digital adaptive equalizer. The digital adaptive equalizer removes inter-symbol interference at the receiver. The digital adaptive equalizer takes unequalized signals from ADC output and uses a combination of feedforward equalizer (FFE) and decision feedback equalizer (DFE) for the best optimized signal-to-noise (SNR) ratio.

11.2.4 Digital Phased Locked Loop (DPLL)

In 100BASE-TX mode, the receive clock is locked to the incoming data stream and extracts a 125 MHz reference clock. The input data stream is quantized by the recovered clock and sent through to the digital adaptive equalizer from each port.

Digital interpolator clock recovery circuits are optimized for MLT-3, NRZI, and Manchester modes. A digital approach makes the device's receiver paths robust to the presence of variations in process, temperature, on-chip noise, and supply voltage.

11.2.5 NRZI to NRZ Conversion

In 100BASE-TX mode, the recovered 100BASE-TX NRZI signal from the receiver is converted to NRZ data, descrambled, aligned, parallelized, and 5B/4B decoded.

11.2.6 Descrambler

The descrambler is initially enabled upon hardware reset if 100BASE-TX is selected. The scrambler can be enabled or disabled via software by setting the DisScrambler bit (Table 170).

The descrambler "locks" to the descrambler state after detecting a sufficient number of consecutive idle code-groups. The receiver does not attempt to decode the data stream unless the descrambler is locked. Once locked, the descrambler continuously monitors the data stream to make sure that it has not lost synchronization.

The receiver descrambles the incoming data stream by exclusive ORing it with the output of an 11-bit wide linear feedback shift register (LFSR), which produces a 2047-bit non-repeating sequence.

The descrambler is always forced into the "unlocked" state when a link failure condition is detected or when insufficient idle symbols are detected.

11.2.7 Serial to Parallel Conversion and 5B/4B Code-Group Alignment

The Serial to Parallel /Symbol Alignment block performs serial to parallel conversion and aligns 5B code-groups to a nibble boundary.

11.2.8 5B/4B Decoder

The 5B/4B decoder translates 5B code-groups into 4B nibbles to be presented to the MAC interfaces. The 5B/4B code mapping is shown in Table 54.

11.2.8.1 FIFO

The 100BASE-X or 10BASE-T packet is placed into the FIFO in order to correct for any clock mismatch between the recovered clock and the reference clock REFCLK.

11.2.8.2 100BASE-FX Receiver

In 100BASE-FX mode, a pseudo-ECL (PECL) receiver is used to decode the incoming NRZI signal passed to the NRZI-NRZ decoder. The NRZI signal from the receiver is converted to NRZ data, aligned, parallelized, and 5B/4B decoded as in the 100BASE-TX mode.

11.2.8.3 Far End Fault Indication (FEFI)

When 100BASE-FX is selected and bit 0 of CONFIG_A is high at hardware reset, then the far end fault detect (FEFD) circuit is enabled. The FEFD enable state can be overridden by programming the DisFEFI bit (Table 170).

Note

The FEFI function is always disabled if 100BASE-TX is selected.

11.2.8.4 10BASE-T Receiver

In 10BASE-T mode, the recovered 10BASE-T signal is decoded from Manchester to NRZ and then aligned. The alignment is necessary to ensure that the start of frame delimiter (SFD) is aligned to the nibble boundary.

In 10BASE-T mode, a receiver is used to decode the differential voltage offset of the Manchester data. Carrier sense is decoded by measuring the magnitude of the voltage offset.

In this mode, the recovered 10BASE-T signal is decoded from Manchester to NRZ data. The data stream is converted from serial to parallel format and aligned. The alignment is necessary to ensure that the start of frame delimiter (SFD) is aligned to a byte or nibble boundary.

For cable lengths greater than 100 meters, the incoming signal has more attenuation. Hence, the receive voltage threshold should be lowered via the Extended Distance bit in the PHY Specific Control Register (Table 170).

Table 54: 5B/4B Code Mapping

PCS Code-Group [4:0] 4 3 2 1 0	Name	TXD/RXD <3:0> 3 2 1 0	Interpretation
1 1 1 1 0	0	0 0 0 0	Data 0
0 1 0 0 1	1	0 0 0 1	Data 1
1 0 1 0 0	2	0 0 1 0	Data 2
1 0 1 0 1	3	0 0 1 1	Data 3
0 1 0 1 0	4	0 1 0 0	Data 4
0 1 0 1 1	5	0 1 0 1	Data 5
0 1 1 1 0	6	0 1 1 0	Data 6
0 1 1 1 1	7	0 1 1 1	Data 7
1 0 0 1 0	8	1 0 0 0	Data 8
1 0 0 1 1	9	1 0 0 1	Data 9
1 0 1 1 0	A	1 0 1 0	Data A
1 0 1 1 1	B	1 0 1 1	Data B
1 1 0 1 0	C	1 1 0 0	Data C
1 1 0 1 1	D	1 1 0 1	Data D
1 1 1 0 0	E	1 1 1 0	Data E
1 1 1 0 1	F	1 1 1 1	Data F
1 1 1 1 1	I	Undefined	IDLE; used as inter-stream fill code
1 1 0 0 0	J	0 1 0 1	Start-of-Stream Delimiter, Part 1 of 2; always used in pairs with K
1 0 0 0 1	K	0 1 0 1	Start-of-Stream Delimiter, Part 2 of 2; always used in pairs with J
0 1 1 0 1	T	Undefined	End-of-Stream Delimiter, Part 1 of 2; always used in pairs with R
0 0 1 1 1	R	Undefined	End-of-Stream Delimiter, Part 2 of 2; always used in pairs with T
0 0 1 0 0	H	Undefined	Transmit Error; used to force signaling errors
0 0 0 0 0	V	Undefined	Invalid code
0 0 0 0 1	V	Undefined	Invalid code
0 0 0 1 0	V	Undefined	Invalid code
0 0 0 1 1	V	Undefined	Invalid code
0 0 1 0 1	V	Undefined	Invalid code
0 0 1 1 0	V	Undefined	Invalid code
0 1 0 0 0	V	Undefined	Invalid code
0 1 1 0 0	V	Undefined	Invalid code
1 0 0 0 0	V	Undefined	Invalid code
1 1 0 0 1	V	Undefined	Invalid code

11.2.9 Setting Cable Characteristics

Since cable characteristics are different between unshielded twisted pair and shielded twisted pair cable, optimal receiver performance can be obtained in 100BASE-TX and 10BASE-T modes by setting the TPSelect bit in the PHY Specific Control Register (Table 170) for cable type.

11.2.10 Scrambler/Descrambler

The scrambler block is initially enabled upon hardware reset if 100BASE-TX is selected. If 100BASE-FX or 10BASE-T is selected, the scrambler is disabled by default. The scrambler is controlled by programming the DisScrambler bit in the PHY Specific Control Register (Table 170).

The scrambler setting is also controlled by hardware configuration at the end of hardware reset. Table 55 shows the effect of various configuration settings on the scrambler.

Table 55: Scrambler Settings

P[1:0]_CONFIG (If FX is selected)	DisScrambler Bit ()	Scrambler/ Descrambler
High	HW reset to 1	Disabled
Low	HW reset to 0	Enabled
X	User set to 1	Disabled
X	User set to 0	Enabled (in copper mode only)

11.2.11 Link Monitor

The link monitor is responsible for determining whether link is established with a link partner.

In 10BASE-T mode, link monitor function is performed by detecting the presence of the valid link pulses on the RXP/N pins.

In 100BASE-TX mode, the link is established by scrambled idles

In 100BASE-FX mode, the external fiber-optic receiver performs the signal detection function and communicates this information with the devices through SDET pins for all individual ports.

If Force Link Good is asserted (ForceLink bit is set high - PHY Control Register, Table 160), the link is forced to be good, and the link monitor is bypassed. Pulse checking is disabled if Auto-Negotiation is disabled, and DisN-LPCheck (PHY Specific Control Register, Table 170) is set high. If Auto-Negotiation is disabled and DisNLPGen (PHY Specific Control Register, Table 170) is set high, then the link pulse transmission is disabled.

11.2.12 Auto-Negotiation

Auto-Negotiation is initiated upon any of the following conditions:

- Power up reset
- Hardware reset
- Software reset (SWReset bit—PHY Control Register, Table 160)
- Restart Auto-Negotiation (RestartAneg bit—PHY Control Register, Table 160)
- Transition from power down to power up (PwrDwn bit—PHY Control Register, Table 160)
- Change from the linkfail state to the link-up state

If Auto-Negotiation is enabled, the devices negotiate with its link partner to determine the speed and duplex mode at which to operate. If the link partner is unable to Auto-Negotiate, the devices go into the parallel detect mode to determine the speed of the link partner. Under parallel detect mode, the duplex mode is fixed at half-duplex.

After hardware reset, Auto-Negotiation can be enabled and disabled via the AnegEn bit (PHY Control Register - Table 160). When Auto-Negotiation is disabled, the speed and duplex can be changed via the SpeedLSB and Duplex bits (PHY Control Register - Table 160), respectively. The abilities that are advertised can be changed via the Auto-Negotiation Advertisement Register (Table 164).

11.2.13 Register Update

Changes to the AnegEn, SpeedLSB, and Duplex bits (Table 160) do not take effect unless one of the following takes place:

- Software reset (SWReset bit - Table 160)
- Restart Auto-Negotiation (RestartAneg bit - Table 160)
- Transition from power down to power up (PwrDwn bit - Table 160)
- Loss of the link

The Auto-Negotiation Advertisement register (Table 164) is internally latched once every time Auto-Negotiation enters the ability detect state in the arbitration state machine. Hence, a write into the Auto-Negotiation Advertisement Register has no effect once the devices begin to transmit Fast Link Pulses (FLPs). This guarantees that a sequence of FLPs transmitted is consistent with one another.

The Next Page Transmit register (Table 168) is internally latched once every time Auto-Negotiation enters the next page exchange state in the arbitration state machine.

11.2.14 Next Page Support

The devices support the use of next page during Auto-Negotiation. By default, the received base page and next page are stored in the Link Partner Ability register - Base Page (Table 165). The devices have an option to write the received next page into the Link Partner Next Page register - Table 169 - (similar to the description provided in the IEEE 802.3ab standard) by programming the Reg8NxtPg bit (PHY Specific Control Register - Table 170).

11.2.15 Status Registers

Once the devices complete Auto-Negotiation, the various statuses in the PHY Status (Table 161), Link Partner Ability (Next Page)(Table 166), and Auto-Negotiation Expansion(Table 167) registers are updated. Speed, duplex, page received, and Auto-Negotiation completed statuses are also available in the PHY Specific Status (Table 171) and PHY Interrupt Status (Table 173) registers.

11.3 Power Management

The devices support advanced power management modes that conserve power. These features are only recommended for power-critical designs only.

11.3.1 Low Power Modes

Two low power modes are supported in the devices.

- IEEE 22.2.4.1.5 compliant power down
- Energy Detect+™

IEEE 22.2.4.1.5 power down compliance allows for the PHY to be placed in a low-power consumption state by register control.

Energy Detect+™ allows the devices to wake up when energy is detected on the wire with the additional capability to wake up a link partner. The 10BASE-T link pulses are sent once every second while listening for energy on the line.

11.3.2 MAC Interface and PHY Configuration for Low Power Modes

The devices have one CONFIG bit dedicated to support the low power modes. Energy Detect may be configured using the CONFIG_B pin, or through PHY Specific Control Register 0x10.

Low power modes are also register programmable. The EDet bit (Table 170) enables the user to turn on Energy Detect+™. When the low power mode is not selected, the PwrDwn bit (Table 160) can be used. If during the energy detect mode, the PHY wakes up and starts operating in normal mode, the EDet bit settings are retained. When the link is lost and energy is no longer detected, the devices return to the mode stored in the EDet bit.

Table 56 shows how these modes are entered

.

Table 56: Operating Mode Power Consumption

Power Mode	Est. Power	How to Activate Mode
IEEE Power down	See Section 15.	PwrDwn bit write (Table 160)
Energy Detect+™	Same as IEEE Power down mode—see Section 15.	Configuration option & register EDet bit write (Table 170)

11.3.3 IEEE Power Down Mode

The standard IEEE power down mode (22.2.4.1.5) is entered by setting the PwrDwn (Table 160) bit equal to one. In this mode, the PHY does not respond to any MAC interface signals except the MDC/MDIO. It also does not respond to any activity on the CAT 5 cable.

In this power down mode, the PHY cannot wake up on its own by detecting activity on the CAT 5 cable. It can only wake up by clearing the PwrDwn bit to 0.

11.3.3.1 Energy Detect +™

In this mode, the PHY sends out a single 10 Mbps NLP (Normal Link Pulse) every one second. If the devices are in Energy Detect+™ mode, it can wake a connected device. When ENA_EDET is 1, the mode of operation is Energy Detect+™. The devices also respond to MDC/MDIO.

11.4 Far End Fault Indication (FEFI)

Far-end fault indication provides a mechanism for transferring information from the local station to the link partner that a remote fault has occurred in 100BASE-FX mode.

A remote fault is an error in the link that one station can detect while the other one cannot. An example of this is a disconnected fiber at a station's transmitter. This station is receiving valid data and detects that the link is good via the link monitor, but is not able to detect that its transmission is not propagating to the other station.

A 100BASE-FX station that detects this remote fault modifies its transmitted idle stream pattern from all ones to a group of 84 ones followed by one zero. This is referred to as the FEFI idle pattern.

The FEFI function is controlled by CONFIG_A connection and the DisFEFI bit (Table 170).

Table 57 shows the various configuration settings affecting the FEFI function on hardware reset.

Table 57: FEFI Select

EN_FEFI Config_A) Conncection	FEFI	DisFEFI Bit (Table 170)
VSS	Disabled	HW reset to 1
P0_LED0	Enabled	HW reset to 0
P0_LED1	Disabled	HW reset to 1
P0_LED2	Enabled	HW reset to 0
P1_LED0	Disabled	HW reset to 1
P1_LED1	Enabled	HW reset to 1
P1_LED2	Disabled	HW reset to 0
VDDO	Enabled	HW reset to 1

11.5 Virtual Cable Tester® Feature

The device's Virtual Cable Tester (VCT™) feature uses Time Domain Reflectometry (TDR) to determine the quality of the cables, connectors, and terminations. Some of the possible problems that can be diagnosed include opens, shorts, cable impedance mismatch, bad connectors, termination mismatch, and bad magnetics.

The devices transmit a signal of known amplitude (+1V) down each of the two pairs of an attached cable. It will conduct the cable diagnostic test on each pair, testing the TX and RX pairs sequentially. The transmitted signal will continue down the cable until it reflects off of a cable imperfection. The magnitude of the reflection and the time it takes for the reflection to come back are shown in the VCT registers (Table 180 and Table 181) on the AmpRfln and DistRfln bits respectively.

Using the information from the VCT registers (Table 180 and Table 181), the distance to the problem location and the type of problem can be determined. For example, the time it takes for the reflection to come back, can be converted to distance using the cable fault distance trend line tables in Table 180 and Figure 78. The polarity and magnitude of the reflection together with the distance will indicate the type of discontinuity. For example, a +1V reflection will indicate an open close to the PHY and a -1V reflection will indicate a short close to the PHY.

When the cable diagnostic feature is activated by setting the ENVCT bit to one (Table 180), a pre-determined amount of time elapses before a test pulse is transmitted. This is to ensure that the link partner loses link, so that it stops sending 100BASE-TX idles or 10 Mbit data packets. This is necessary to be able to perform the TDR test. The TDR test can be performed either when there is no link partner or when the link partner is Auto-Negotiating or sending 10 Mbit idle link pulses. If the devices receive a continuous signal for 125 ms, it will declare test failure because it cannot start the TDR test. In the test fail case, the received data is not valid. The results of the test are also summarized in the VCTTst bits (Table 180 and Table 181).

- 11 = Test fail (The TDR test could not be run for reasons explained above)
- 00 = valid test, normal cable (no short or open in cable)
- 10 = valid test, open in cable (Impedance > 333 ohms)
- 01 = valid test, short in cable (Impedance < 33 ohms)

The definition for shorts and opens is arbitrary and the user can define it anyway they desire using the information in the VCT registers (Table 180 and Table 181). The impedance mismatch at the location of the discontinuity could also be calculated knowing the magnitude of the reflection. Refer to the App Note "Virtual Cable Tester -- How to use TDR results" for details.

11.6 Data Terminal Equipment (DTE) Detect

The devices support the Data Terminal Equipment (DTE) detect function. The IEEE 802.3af - 2003 DTE power scheme requires no role of the PHY to detect a DTE link partner that requires power; however, the devices offer the ability to detect a DTE link partner that requires power.

The DTE power function can be enabled after a hardware reset by writing to the Enable DTE Detect bit (Register 16.15), followed by a software reset. When DTE Detect is enabled, the devices will first monitor for any activity transmitted by the link partner. If the link partner is active, then the link partner has power and can link with the device. If there is no activity coming from the link partner, DTE Detect engages, and special pulses are sent to detect if the link partner requires DTE power. If the link partner has a low pass filter (or similar fixture) installed, the link partner will be detected as requiring DTE power.

The detection of the DTE power requirement can be reported to the devices in two ways.

- The DTE Detect Status bit (Register 17.15) immediately asserts as soon as the link partner is detected as a device requiring DTE power. The devices will continually send special link pulses to detect if the link partner requires DTE power.
- If the DTE Detect Status Change Interrupt bit (Register 18.15) is enabled, an interrupt can be generated if a DTE powered device is detected. The devices will then update the DTE Detect Status Drop bit (Register 22.15) and the DTE Detect Status bit (Register 17.15).

If a link partner that requires DTE power is unplugged, the DTE Detect Status bit (Register 17.15) will drop after a user controlled delay (default is 20 seconds - DTE Detect Status Drop bit (Register 22.15:14) = 0x04), since the lowpass filter (or similar fixture) is removed during power up.

A detailed description of the register bits used for DTE power detection for sthe device is shown in Table 58.

Note

Auto-Negotiation must be enabled to use DTE detect. The DTE detect must be turned off when performing the Virtual Cable Tester® (VCT™) test.

Table 58: Registers for DTE Detect

Register	Description
16.15 - Enable DTE Detect	1 = Enable DTE detect 0 = Disable DTE detect Default at HW reset: 0 At SW reset: Retain
17.15 DTE Detect status	1 = DTE detected 0 = DTE not detected Default at HW reset: 0 At SW reset: 0
18.15 - DTE Detect state changed interrupt enable	1 = Interrupt enable 0 = Interrupt disable Default at HW reset: 0 At SW reset: retain
19.15 DTE Detect state changed interrupt	1 = Changed 0 = No change Default at HW reset: 0 At SW reset: 0
22.15:12 DTE detect status drop	Once the devices no longer detect the link partner's DTE filter, the devices will wait a period of time before clearing the DTE detection status bit (17.15). The wait time is 5 seconds multiplied by the value of these bits. Default at HW reset: 0x4 At SW reset: retain

11.7 Auto MDI/MDIX Crossover

The devices automatically determine whether or not it needs to cross over between pairs so that an external crossover cable is not required. If the devices interoperate with a device that cannot automatically correct for crossover, the devices make the necessary adjustment prior to commencing Auto-Negotiation. If the devices interoperate with a device that implements MDI/MDIX crossover, a random algorithm as described in IEEE 802.3 Section 40.4.4 determines which device performs the crossover.

When the devices interoperate with legacy 10BASE-T devices that do not implement Auto-Negotiation, the devices follow the same algorithm as described above since link pulses are present. However, when interoperating with legacy 100BASE-TX devices that do not implement Auto-Negotiation (i.e. link pulses are not present), the devices use signal detect to determine whether or not to crossover.

The Auto MDI/MDIX crossover function can be disabled via the AutoMDI[X] bits (Table 170).

The devices are set to MDIX mode by default if auto MDI/MDIX crossover is disabled at hardware reset.

The pin mapping in MDI and MDIX modes is specified in Table 59.

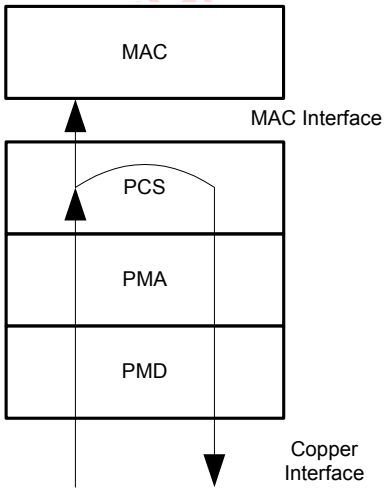
Table 59: MDI/MDIX Pin Functions

Physical Pin	MDIX		MDI	
	100BASE-TX	10BASE-T	100BASE-TX	10BASE-T
TXP/TXN	Transmit	Transmit	Receive	Receive
RXP/RXN	Receive	Receive	Transmit	Transmit

11.8 Copper Line Loopback

Line loopback allows a link partner to send frames into the devices to test the transmit and receive data path. Frames from a link partner into the PHY, before reaching the MAC interface pins, are looped back and sent out on the copper line. They are also sent to the MAC. The packets received from the MAC are ignored during copper line loopback. Refer to Figure 65. This allows the link partner to receive its own frames.

Figure 65: Line Loopback Data Path



Before enabling the line loopback feature, the PHY must first establish link to another PHY link partner. If Auto-Negotiation is enabled, both link partners should advertise the same speed and full-duplex. If Auto-Negotiation is disabled, both link partners need to be forced to the same speed and full-duplex. Once link is established, enable the line loopback mode by writing to Register 28.4.

Line loopback mode works for 10BASE-T, 100BASE-TX, and 100BASE-FX modes.

- 28.4 = 1 (Enable line loopback)
- 28.4 = 0 (Disable line loopback)

11.9 LED Interface

The LEDs can either be controlled by the PHY or controlled externally, independent of the state of the PHY.

External control is achieved by writing to the PHY Manual LED Override register (Table 179). Any of the LEDs can be turned on, off, or made to blink[1] at variable rates independent of the state of the PHY. This independence eliminates the need for driving LEDs from the MAC or the CPU. If the LEDs are driven from the CPU located at the back of the board, the LED lines crossing the entire board will pick up noise. This noise will cause EMI issues. Also, PCB layout will be more difficult due to the additional lines routed across the board.

When the LEDs are controlled by the PHY, the activity of the LEDs is determined by the state of the PHY. Each LED can be programmed to indicate various PHY states, with variable blink rate.

Any one of the LEDs can be controlled independently of the other LEDs (i.e one LED can be externally controlled while another LED can be controlled by the state of the PHY).

The LED interface supports a 3-pin parallel interface for each port or a serial interface for all ports.

The devices can be programmed to display serial LED statuses in single or dual LED modes. Some of the statuses can be pulse stretched. Pulse stretching is necessary because the duration of these status events might be too short to be observable on the LEDs. The pulse stretch duration can be programmed via the PulseStretch bits (Table 178). The default pulse stretch duration is set to 170 to 340 ms. The pulse stretch duration applies to all applicable LEDs.

Some of the statuses indicate multiple events by blinking LEDs. The blink period can be programmed via the BlinkRate bits (Table 178). The default blink period is set to 84 ms. The blink rate applies to all applicable LEDs.

11.9.1 Parallel LED Interface

The parallel LED interface displays 3 different statuses for each port. LED2, LED1, and LED0 pins are used for each port. The LED Parallel Select Register specifies which single LED mode status to display on the LED pins. The defaults to display shown in Table 176 are based on the LED_DEF[1] and LED_DEF[0] values during hardware configuration through the CONFIG_A pin—see Table 4.

Table 60: Parallel LED Hardware Defaults

LED Mode —set by CONFIG_A at reset	P[7:0]_LED2	P[7:0]_LED1	P[7:0]_LED0
0	LINK	RX	TX
1	LINK	ACT	SPEED
2	LINK/RX	TX	SPEED
3	LINK/ACT	DUPLEX/COLX	SPEED

Table 176 shows additional display modes that can be set up by software after startup. Table 61 includes these extra modes that cannot be set up by hardware configuration pins at reset.

1. Energy Detect (Section 11.3) must be disabled on ports that are configured to blink an LED but don't have a link established.

Table 61: Parallel LED Display Interpretation

Status	Description
COLX	Low = Collision activity High = No collision activity This status is pulse stretched to 170 ms.
ERROR	Low = Jabber, received error, false carrier, or FIFO over/underflow occurred High = None of the above occurred This status is pulse stretched to 170 ms.
DUPLEX	Low = Full-duplex High = Half-duplex
LINK	Low = Link up High = Link down
RX	Low = Receive activity High = No receive activity This status is pulse stretched to 170 ms.
TX	Low = Transmit activity High = No transmit activity This status is pulse stretched to 170 ms.
ACT	Low = Transmit or received activity High = No transmit or receive activity This status is pulse stretched to 170 ms.
SPEED	Low = Speed is 100 Mbps High = Speed is 10 Mbps
LINK/RX	Low = Link up High = Link down Blink = Receive activity (blink rate is 84 ms active then 84ms inactive) The receive activity is pulse stretched to 84 ms.
LINK/ACT	Low = Link up High = Link down Blink = transmit or receive activity (blink rate is 84 ms active then 84 ms inactive) The transmit and receive activity is pulse stretched to 84 ms
DUPLEX/COLX	Low = Full-duplex High = Half-duplex Blink = collision activity (blink rate is 84 ms active then 84 ms inactive) The collision activity is pulse stretched to 84 ms.
ACT (blink mode)	Low = Full-duplex High = Half-duplex Blink = Collision activity (blink rate is 84 ms active then 84 ms inactive) The collision activity is pulse stretched to 84 ms.

11.9.2 Using Two Color LEDs

Two color LEDs are supported with the parallel LED pins on the devices. Each Px_LED[2:0] pin can be independently controlled to be active low (the default) or active high. They can also be independently selected to be active only if the port is in 10 Mbps mode or only 100 Mbps mode or either speed (the default). This can be configured by using the PHY Manual LED override register (Table 179).

The following lists some possible applications of two color LEDs:

Case One:

- Off - No Link
- Green - Link at 100 Mbps
- Yellow - Link at 10 Mbps
- Blink Green - Activity at 100 Mbps
- Blink Yellow - Activity at 10 Mbps

Case Two:

- Off - No Link
- Green - Link, or VCT™ Good
- Blink Green - Activity
- Red - VCT Short
- Blink Red - Testing VCT

Examples of each of these cases follow.

Figure 66: Possible Solutions for Case One

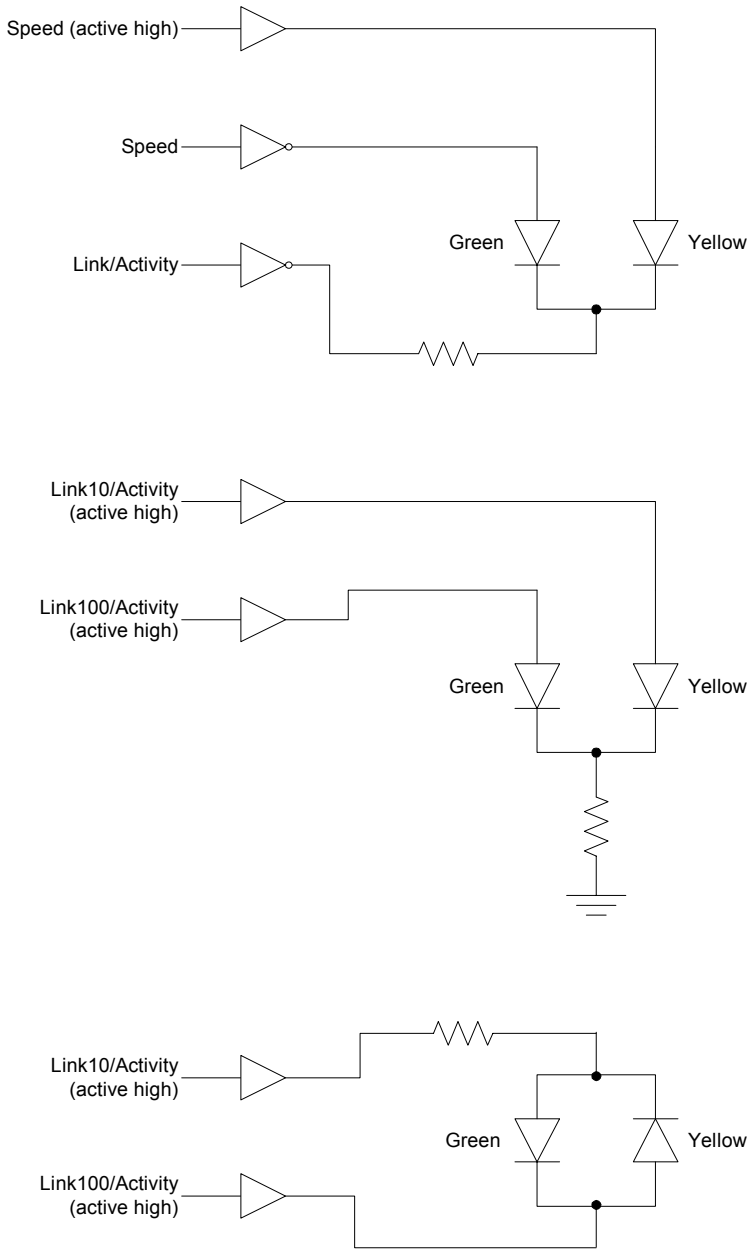
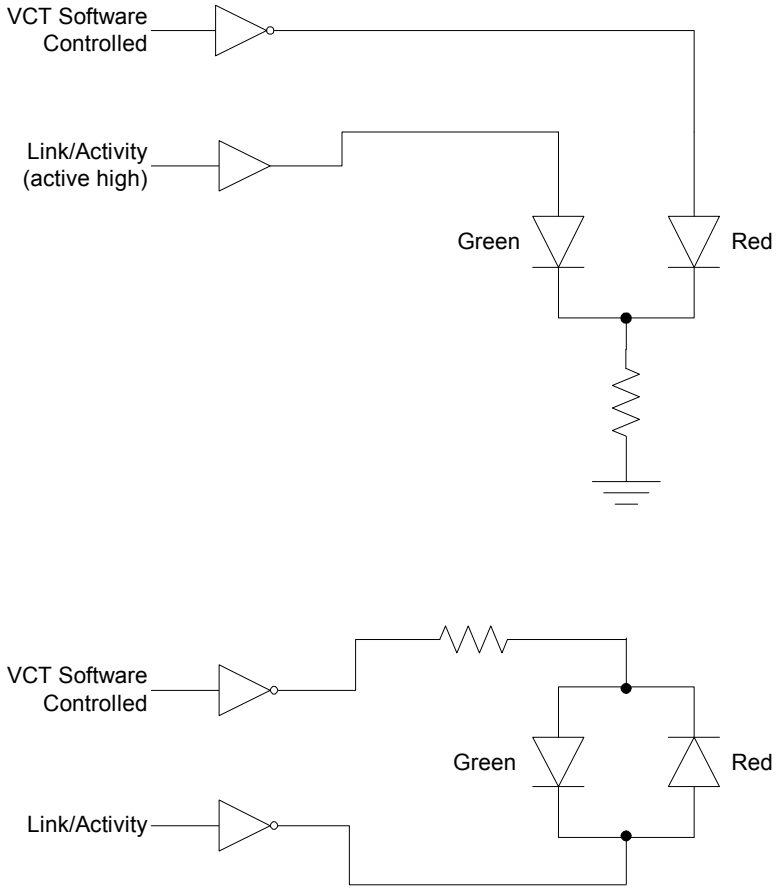


Figure 67: Possible Solutions for Case Two



11.9.3 Serial LED Interface (88E6097F Device Only)

LEDSER, LEDENA, and LEDCLK pins of the 88E6097F device are used for the serial interface. The CONFIG_A pin is used to select 1 of 4 possible modes. The serial LED interface can display up to 11 different statuses in 100BASE-TX and 10BASE-T modes. Statuses to display, pulse stretching, and blink mode can be programmed via the LED Stream Select for Serial LEDs register and the PHY LED Control register bits 5:0.

11.9.4 Single and Dual LED Modes

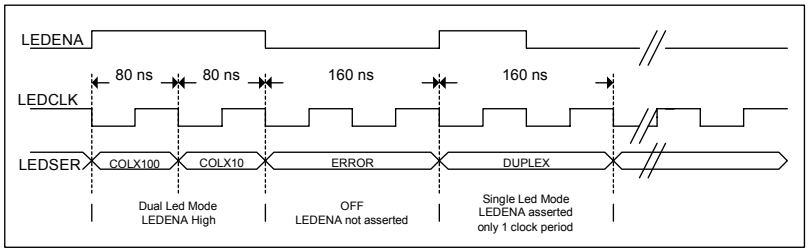
11.9.4.1 Single LED Display Mode

In the single LED display mode, the same status is driven on both status 100 and status 10 positions in the bit stream. However, the LEDENA signal asserts only over the status that is set and de-asserts over the other position that is turned off in the bit stream. For example, DUPLEX shows the same status for DUPLEX100 and DUPLEX10. However, LEDENA signal is high over Duplex100 position only for one clock period. Refer to Figure 68 for more details.

11.9.4.2 Dual LED Display Mode

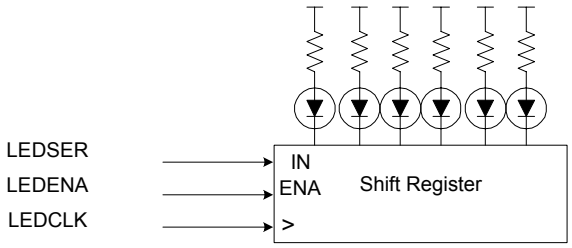
In the dual LED display mode, two LEDs are used: one for 10 Mbps, and one for 100 Mbps activity. A different status is driven on status 100 and status 10 positions in the bit stream. In this case, the LEDENA signal asserts over both 100 and 10 positions in the bit stream. For example, LEDENA signal asserts over COLX100 and COLX10 in Figure 68. LEDENA signal is high for two clock periods. If a particular status bit is turned off, then LEDENA is not asserted in both positions. Figure 68 illustrates single and dual LED modes.

Figure 68: Serial LEDENA High Clocking with COLX in Dual Mode, Error Off, and DUPLEX in Single Mode



The bit stream on LEDSER can be clocked into a shift register with LEDENA as the shift enable signal as shown in Figure 69. The rate of update of the serial LED interface is controlled by programming the PHY LED Control Register bits 8:6. The default value is set to 42 ms.

Figure 69: Serial LED Conversion



After the LED data is shifted into the correct position, the shift sequence is suspended to allow the appropriate LEDs to light or extinguish depending on status. The LED implementation used in the 88E6097F device is self-synchronizing. The default display options are given in Table 62.

Table 62: Serial LED Hardware Defaults (S = Single)

LED_DEF[1] (At Reset)	LED_DEF[0] (At Reset)	COLX	ERROR	DUPLEX	DUPLEX/ COLX	SPEED	LINK	TX	RX	ACT	LINK/RX	LINK/ACT
0	0	Off	S	Off	S	S	Off	Off	Off	Off	Off	S
0	1	S	S	S	Off	S	Off	S	Off	Off	S	Off
1	0	S	S	S	Off	S	S	Off	Off	S	Off	Off
1	1	S	S	S	Off	S	S	S	S	Off	Off	Off

The LED status bits are output in the order shown on the LEDSER pin synchronously to LEDCLK. All statuses for Port 0 are sent out first followed by those for Ports 1 through 7. Each bit in the stream occupies a period of 80 ns.

Table 63: Serial LED Display Order

<COLX100> → <COLX10> → <ERROR100> → <ERROR10> → <DUPLEX100> → <DUPLEX10>→ <DUPLEX/COLX100> → <DUPLEX/COLX10> → <SPEED100> → <SPEED10> → <LINK100>→ <LINK10> → <TX100> → <TX10> → <RX100> → <RX10> → <ACT100> → <ACT10> → <LINK/RX100>→ <LINK/RX10> → <LINK/ACT100> → <LINK/ACT10>

The following tables show the status events that can be displayed by programming the device in single and dual LED display modes

Table 64: Single LED Display Mode

Status	Description
COLX	Low = Collision activity High = No collision activity This status has a default pulse stretch duration of 170 ms.
ERROR	Low = Jabber, received error, false carrier, or FIFO over/underflow occurred High = None of the above occurred This status has a default pulse stretch duration of 170 ms.
DUPLEX	Low = Full-duplex High = Half-duplex
DUPLEX/COLX	Low = Full-duplex High = Half-duplex Blink = Collision activity (default blink rate is 84 ms active then 84 ms inactive) The collision activity has a default pulse stretch duration of 84 ms.
SPEED	Low = Speed is 100 Mbps High = Speed is 10 Mbps
LINK	Low = Link up High = Link down
TX	Low = Transmit activity High = No transmit activity This status has a default pulse stretch duration of 170 ms.
RX	Low = Receive activity High = No receive activity This status has a default pulse stretch duration of 170 ms.
ACT	Low = Transmit or received activity High = No transmit or receive activity This status has a default pulse stretch duration of 170 ms.
LINK/RX	Low = Link up High = Link down Blink = receive activity (blink rate is 84 ms active then 84 ms inactive) The receive activity has a pulse stretch duration of 84 ms.
LINK/ACT	Low = Link up High = Link down Blink = Transmit or receive activity (blink rate is 84 ms active then 84 ms inactive) The transmit and receive activity has a pulse stretch duration of 84 ms.

Table 65: Dual LED Display Mode

Event	Status
COLX100	1 = No 100 Mbps collision activity 0 = 100 Mbps collision activity This status has a default pulse stretch duration of 170 ms.
COLX10	1 = No 10 Mbps collision activity 0 = 10 Mbps collision activity This status has a default pulse stretch duration of 170 ms.
ERROR100	1 = None of the above occurred 0 = Received error, false carrier, or 100 Mbps FIFO over/underflow occurred This status has a default pulse stretch duration of 170 ms.
ERROR10	1 = None of the above occurred 0 = Jabber or 10 Mbps FIFO over/underflow occurred This status has a default pulse stretch duration of 170 ms.
DUPLEX100	1 = 100 Mbps half-duplex 0 = 100 Mbps full-duplex
DUPLEX10	1 = 10 Mbps half-duplex 0 = 10 Mbps full-duplex
DUPLEX/COLX100	1 = 100 Mbps half-duplex 0 = 100 Mbps full-duplex Blink = 100 Mbps collision activity The 100 Mbps collision activity has a default pulse stretch duration of 84 ms. The blink rate can be programmed.
DUPLEX/COLX10	1 = 10 Mbps half-duplex 0 = 10 Mbps full-duplex Blink = 10 Mbps collision activity The 10 Mbps collision activity has a default pulse stretch duration of 84 ms. The blink rate can be programmed.
SPEED100	1 = Speed is 10 Mbps 0 = Speed is 100 Mbps
SPEED10	1 = Speed is 100 Mbps 0 = Speed is 10 Mbps
LINK100	1 = 100 Mbps link down 0 = 100 Mbps link up
LINK10	1 = 10 Mbps link down 0 = 10 Mbps link up
TX100	1 = No 100 Mbps transmit activity 0 = 100 Mbps transmit activity This status has a default pulse stretch duration of 170 ms.

Table 65: Dual LED Display Mode (Continued)

Event	Status
TX10	1 = No 10 Mbps transmit activity 0 = 10 Mbps transmit activity This status has a default pulse stretch duration of 170 ms.
RX100	1 = No 100 Mbps receive activity 0 = 100 Mbps receive activity This status has a default pulse stretch duration of 170 ms.
RX10	1 = No 10 Mbps receive activity 0 = 10 Mbps receive activity This status has a default pulse stretch duration of 170 ms.
ACT100	1 = No 100 Mbps transmit or 100 Mbps receive activity 0 = 100 Mbps transmit or 100 Mbps receive activity This status has a default pulse stretch duration of 170 ms.
ACT10	1 = No 10 Mbps transmit or 10 Mbps receive activity 0 = 10 Mbps transmit or 10 Mbps receive activity This status has a default pulse stretch duration of 170 ms.
LINK/RX100	1 = 100 Mbps link down 0 = 100 Mbps link up Blink = 100 Mbps receive activity The 100 Mbps receive activity has a default pulse stretch duration of 84 ms. The blink rate can be programmed.
LINK/RX10	1 = 10 Mbps link down 0 = 10 Mbps link up Blink = 10 Mbps receive activity The 10 Mbps receive activity has a default pulse stretch duration of 84 ms. The blink rate can be programmed.
LINK/ACT100	1 = 100 Mbps link down 0 = 100 Mbps link up Blink = 100 Mbps transmit or 100 Mbps receive activity The 100 Mbps receive or transmit activity has a default pulse stretch duration of 84 ms. The blink rate can be programmed.
LINK/ACT10	1 = 10 Mbps link down 0 = 10 Mbps link up Blink = 10 Mbps transmit or 10 Mbps receive activity This 10 Mbps receive or transmit activity has a default pulse stretch duration of 84 ms. The blink rate can be programmed.

Section 12. Serial Management Interface (SMI)

The device's serial management interface provides access to the internal registers via the MDC and MDIO signals and is compliant with IEEE 802.3u clause 22. MDC is the management data clock input whose frequency can run from DC to a maximum rate of 8.3 MHz. MDIO is the management data input/output which carries a bidirectional signal that runs synchronously with the MDC. The MDIO pin requires a pull-up resistor to pull the MDIO high during idle and turnaround times.

12.1 MDC/MDIO Read and Write Operations

All of the relevant serial management registers, as well as several optional registers, are implemented in the device's Switch Core. A description of these registers can be found in Section 13. "Switch Register Description" .

Figure 70: Typical MDC/MDIO Read Operation

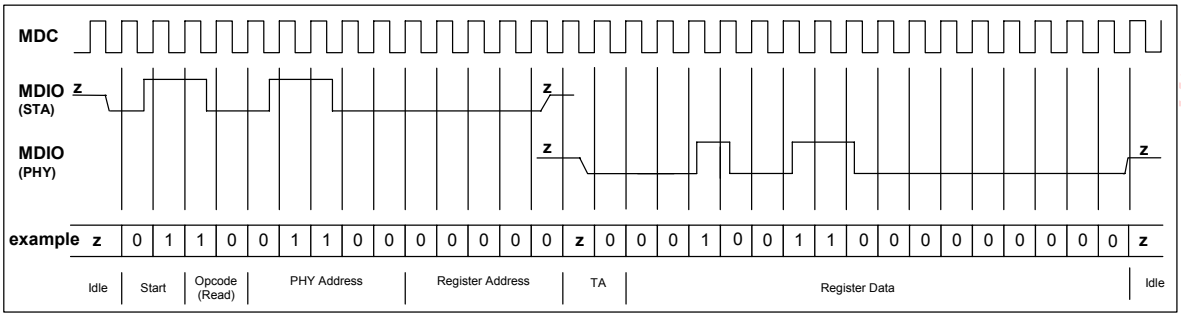


Figure 71: Typical MDC/MDIO Write Operation

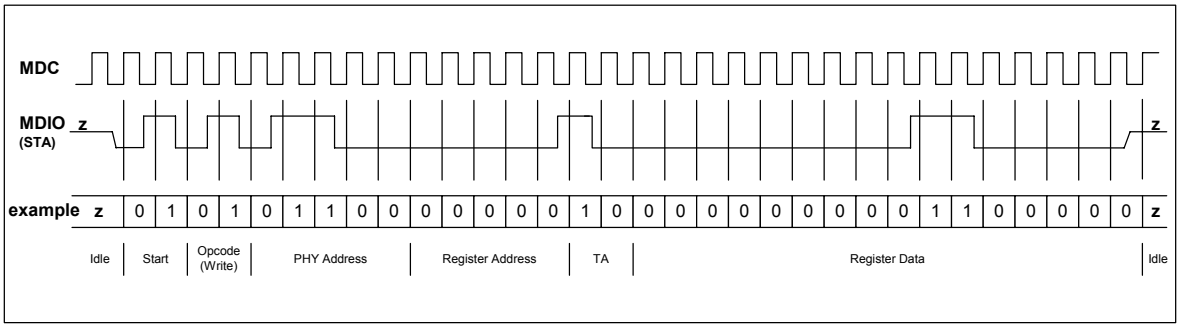


Table 66 shows an example of a read operation of PHY address 04, register 0, with data of 04C0.

Table 66: Serial Management Interface Protocol Example

32-Bit Preamble	Start of Frame	Opcode Read = 10 Write = 01	5-Bit Phy Device Address	5-Bit Phy Register Address	2-Bit Turnaround Read = z0 Write = 10	16-Bit Data Field	Idle
11111111	01	10	00100	00000	z0	0000010011000000	11111111

Section 13. Switch Register Description

The devices registers are accessible using the MDC_CPU and MDIO_CPU pins which support the IEEE Serial Management Interface (SMI – Clause 22) used for PHY devices (Refer to MDC/MDIO in Table 15) or by using Ethernet frames using Remote Management (Section 10.). The devices support two kinds of SMI address usage models. One uses 1 of the 32 possible Device addresses (when in Multi-chip mode - 13.2 "Multi-chip Addressing Mode"). The other uses all of the 32 possible Device addresses (when in Single-chip mode - 13.3 "Single-chip Addressing Mode"). The device addresses and modes used are configurable at reset with the ADDR[4:0] configuration pins (Refer to "GMII/MII Transmit Interface," Table 8 and Table 10).

Table 67: Register Map—Multi-Chip Addressing Mode

Address	Description	Page #
00	"SMI Command Register"	page 242
01	"SMI Data Register"	page 242

Table 68: Register Map—Single-Chip Addressing Mode

Address	Description	Page #
Switch Per-port Registers		
00	"Port Status Register"	page 246
01	"PCS Control Register"	page 250
02	"Jamming Control Register"	page 252
03	"Switch Identifier Register"	page 253
04	"Port Control Register"	page 254
05	"Port Control 1"	page 260
06	"Port Based VLAN Map"	page 261
07	"Default Port VLAN ID & Priority,"	page 262
08	"Port Control 2 Register"	page 263
09	"Egress Rate Control"	page 266
10	"Egress Rate Control 2"	page 267
11	"Port Association Vector"	page 269
12	"Port ATU Control"	page 271
13	"Priority Override Register"	page 273
14	"Policy Control Register"	page 276
15	"Port E Type"	page 280
16	"InDiscards Low Counter"	page 281

Table 68: Register Map—Single-Chip Addressing Mode (Continued)

Address	Description	Page #
17	"InDiscards High Counter"	page 281
18	"InFiltered Counter"	page 281
19	"OutFiltered Counter"	page 282
20–23	Reserved	
24	"Port IEEE Priority Remapping Registers"	page 283
25	"Port IEEE Priority Remapping Registers"	page 284
26	"RGMII Timing Control (Device Offset 0x16 only)"	page 285
27	"Queue Counter Registers"	page 285
28-31	Reserved	
Switch Global Registers		
00	"Switch Global Status Register"	page 287
01	"ATU FID Register"	page 288
02	"VTU FID Register"	page 289
03	"VTU SID Register"	page 289
04	"Switch Global Control Register"	page 290
05	"VTU Operation Register"	page 291
06	"VTU VID Register"	page 293
07	"VTU/STU Data Register Ports 0 to 3 for VTU Operations"	page 294
	"VTU/STU Data Register Ports 0 to 3 for STU Operations"	page 295
08	"VTU/STU Data Register Ports 4 to 7 for VTU Operations"	page 296
	"VTU/STU Data Register Ports 4 to 7 for STU Operations"	page 297
09	"VTU/STU Data Register Port 8 to 10 for VTU Operations"	page 297
	"VTU/STU Data Register Port 8 to 10 for STU Operations"	page 298
10	"ATU Control Register"	page 299
11	"ATU Operation Register"	page 300
12	"ATU Data Register"	page 302
13	"ATU MAC Address Register Bytes 0 & 1"	page 303
14	"ATU MAC Address Register Bytes 2 & 3"	page 303
15	"ATU MAC Address Register Bytes 4 & 5"	page 303
16	"IP-PRI Mapping Register 0"	page 304
17	"IP-PRI Mapping Register 1"	page 304

Table 68: Register Map—Single-Chip Addressing Mode (Continued)

Address	Description	Page #
18	"IP-PRI Mapping Register 2"	page 305
19	"IP-PRI Mapping Register 3"	page 305
20	"IP-PRI Mapping Register 4"	page 306
21	"IP-PRI Mapping Register 5"	page 306
22	"IP-PRI Mapping Register 6"	page 307
23	"IP-PRI Mapping Register 7"	page 307
24	"IEEE-PRI Register"	page 308
25	Reserved	
26	"Monitor Control"	page 309
27	"Total Free Counter"	page 311
28	"Global Control 2"	page 312
29	"Stats Operation Register"	page 313
30	"Stats Counter Register Bytes 3 & 2"	page 315
31	"Stats Counter Register Bytes 1 & 0"	page 315
Switch Global 2 Registers		
01	"Interrupt Source Register"	page 317
02	"Interrupt Mask Register"	page 317
02	"MGMT Enable Register 2x"	page 318
03	"MGMT Enable Register 0x"	page 318
04	"Flow Control Delay Register"	page 319
05	"Switch Management Register"	page 320
06	"Device Mapping Table Register"	page 322
07	"Trunk Mask Table Register"	page 323
08	"Trunk Members Table Register"	page 323
09	"Ingress Rate Command Register"	page 324
10	"Ingress Rate Data Register"	page 324
11	"Cross-chip Port VLAN Register"	page 325
12	"Cross-chip Port VLAN Data Register"	page 326
13	"Switch MAC Register"	page 327
14	"ATU Stats Register"	page 328
15	"Priority Override Table"	page 329
16-23	Reserved	
24	"SMI PHY Command Register"	page 332
25	"SMI PHY Data Register"	page 332

Table 68: Register Map—Single-Chip Addressing Mode (Continued)

Address	Description	Page #
26	Reserved	
27	"Watch Dog Control Register"	page 333
28	"QoS Weights Register"	page 335
29	"SDET Polarity Register"	page 336
PIRL Registers		
0	"PIRL Bucket Configuration Register"	page 338
1	"PIRL Bucket Configuration Register"	page 339
2	"PIRL Bucket Configuration Register"	page 339
3	"PIRL Bucket Configuration Register"	page 340
4	"PIRL Bucket Configuration Register"	page 341
5	"PIRL Bucket Configuration Register"	page 341
6	"PIRL Bucket Configuration Register"	page 342
7	"PIRL Bucket Configuration Register"	page 344

13.1 Register Types

The registers in the devices are made up of one or more fields. The way in which each of these fields operate is defined by the field's Type. The function of each Type is described below.

Type	Description
LH	Register field with latching high function. If status is high, then the register bit is set to one and remains set until a read operation is performed through the management interface or a reset occurs.
LL	Register field with latching low function. If status is low, then the register bit is cleared to zero and remains zero until a read operation is performed through the management interface or a reset occurs.
RES	Reserved for future use. All reserved bits are read as zero unless otherwise noted.
RO	Read only.
R/W	Read and write with initial value indicated.
RWR	Read/Write reset. All field bits are readable and writable. After reset, register field is cleared to zero.
RWS	Read/Write set. All field bits are readable and writable. After reset, register field is set to a non-zero value specified in the text.
SC	Self-Clear. Writing a one to this register causes the desired function to be immediately executed, then the register field is automatically cleared to zero when the function is complete.
Update	Value written to the register field does not take effect until soft reset is executed; however the written value can be read even before the software reset.
Retain	Value written to the register does not take effect without a software reset and the computer maintains its value after a software reset.
WBAR	Write back as read. All fields must be read and left unchanged before performing a write.
WO	Write only. Reads to this type of register field return undefined data.

13.2 Multi-chip Addressing Mode

When Multi-chip addressing mode is used on the SMI interface, the devices respond to only 1 of the 32 possible SMI device addresses so that it can share the SMI interface with multiple devices. In this mode, only two of the devices' registers are directly accessible, the SMI Command register (Table 69) and the SMI Data register (Table 70). These two registers are used to access all the other device registers indirectly (along with any PHY registers that may be attached to it).

Indirect accessing of the other devices registers is accomplished by setting the SMI Command register's DevAddr and RegAddr bits to point to the devices register to access. Use the DevAddr and RegAddr values defined for devices in the Single-chip Addressing mode (Section 13.3).

Multi-chip Addressing Mode is enabled when the ADDR[4:0] configuration pins (Table 10) carry a non-zero value at the rising edge of RESETn. The ADDR[4:0] configuration pins also define the single SMI address to which this device will respond to. To avoid conflicts, this requires that all devices on the same SMI interface use unique ADDR[4:0] values. An SMI address of 0x00 is not supported in this mode as this value on the ADDR[4:0] pins places the devices into Single-chip Addressing mode (Section 13.3).

Table 69: SMI Command Register[1]
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	SMIBusy	SC	Internal SMI Unit Busy. This bit must be set to a one to start an internal SMI operation (see SMIOp below). Only one SMI operation can be executing at one time so this bit must be zero before setting it to a one. When the requested SMI operation is completed, this bit is automatically be cleared to a zero.
14:13	Reserved	RES	Reserved for Future Use
12	SMIMode	RWR	Internal SMI Mode bit. This bit is used to define the SMI frame type to generate as follows: 0 = Generate IEEE 802.3 Clause 45 SMI frames[2] 1 = Generate IEEE 802.3 Clause 22 SMI frames[3]
11:10	SMIOp	RWR	Internal SMI Opcode. These bits are used to select the SMI opcode to operate on during SMI commands as follows: When the SMIMode bit = 1 then SMIOp = (IEEE 802.3 Clause 22): 11 = Reserved 10 = Read Data Register 01 = Write Data Register 00 = Reserved When the SMIMode bit = 0 then SMIOp = (IEEE 802.3 Clause 45): 11 = Read Data Register with post increment on the address register 10 = Read Data Register 01 = Write Data Register 00 = Write Address Register
9:5	DevAddr	RWR	Internal SMI Device Address bits. These bits are used to select the SMI device (Clause 22) or port (Clause 45) to operate on during SMI commands.
4:0	RegAddr	RWR	Internal SMI Register Address bits. These bits are used to select the SMI register (Clause 22) or device class (Clause 45) to operate on during SMI commands.

1. This register is accessible only when the device is in Multi-chip Addressing mode.
2. Clause 45 SMI frames can be used to access Clause 45 devices connected to the MDC_PHY and MDIO_PHY pins
3. Clause 22 SMI frames must be used to access the internal switch registers

Table 70: SMI Data Register[1]
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15:0	SMIData	RWR	SMI Data register. During SMI writes these bits must be written with the SMI data to be written prior to starting the SMI operation (i.e., before setting SMIBusy to a one). During SMI reads these bits will contain the SMI data that was read after the SMI read operation is completed (i.e., SMIBusy returns to a zero). Writes to this register must not be done while SMIBusy is a one.

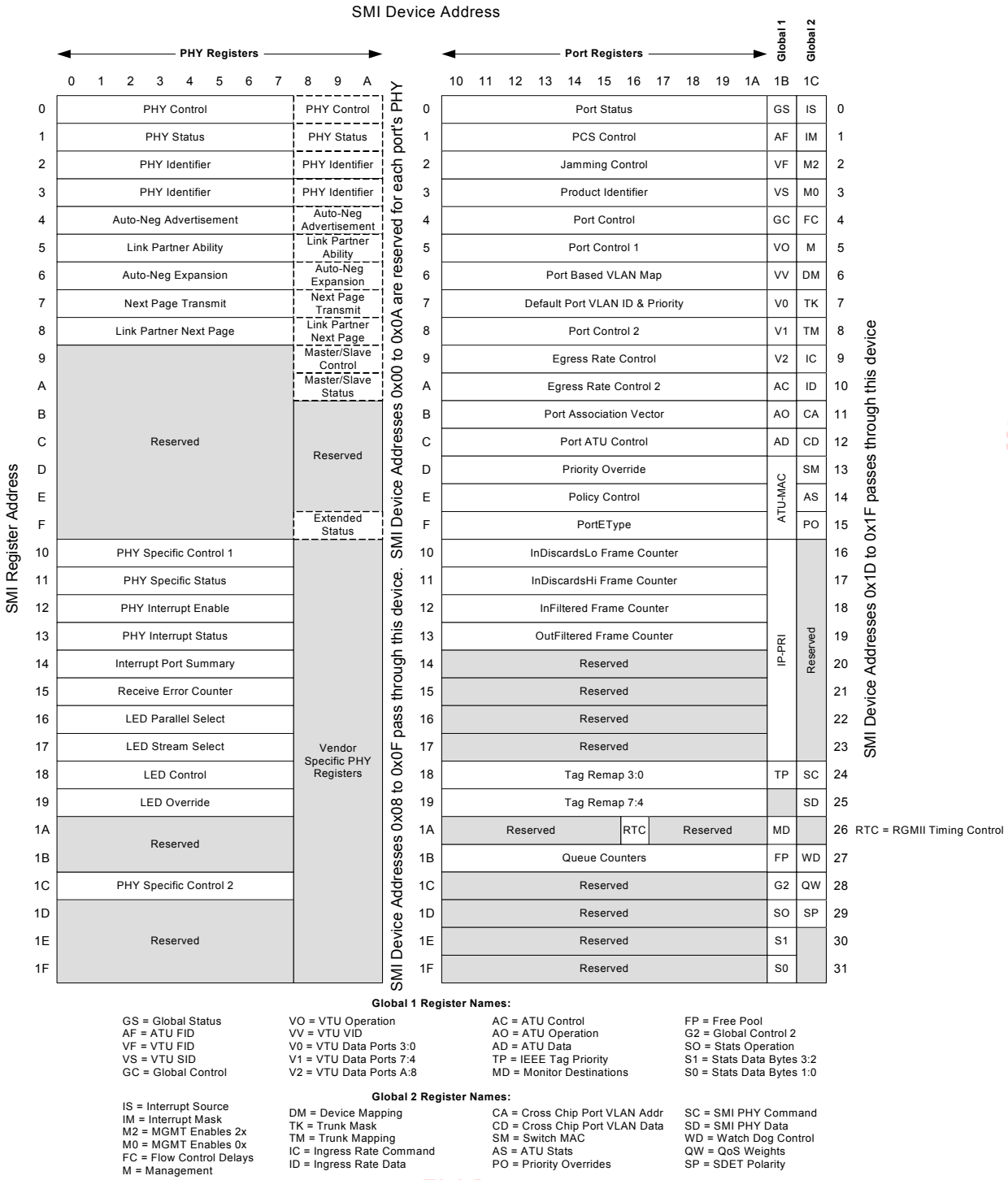
1. This register is accessible only when the device is in Multi-chip Addressing mode.

13.3 Single-chip Addressing Mode

Figure 72 shows the register map in Single-chip mode and for Remote Management register accesses (Section 10.5.5). In this mode, the devices respond to all 32 SMI device addresses so it must be the only device connected to a SMI Master (typically a CPU). The devices use 21 SMI device addresses internally (0x00 to 0x07 and 0x10 to 0x1C) and all unused SMI device addresses are ignored. If PHYs are attached to the devices Ports 8, 9, or 10, they must use SMI device addresses 0x08 (for Port 8) to 0x0A (for Port 10) so that the PPU can automatically poll the PHYs for link, speed, duplex and flow control status, and communicate the current PHY state to the correct MAC.

Single-chip Addressing Mode is enabled when the ADDR[4:0] configuration pins (Table 10) are all zeroes at the rising edge of RESETn.

Figure 72: Device Register Map



Global 1 Register Names:

GS = Global Status
AF = ATU FID
VF = VTU FID
VS = VTU SID
GC = Global Control

VO = VTU Operation
VV = VTU VID
V0 = VTU Data Ports 3:0
V1 = VTU Data Ports 7:4
V2 = VTU Data Ports A:8

AC = ATU Control
AO = ATU Operation
AD = ATU Data
TP = IEEE Tag Priority
MD = Monitor Destinations

FP = Free Pool
G2 = Global Control 2
SO = Stats Operation
S1 = Stats Data Bytes 3:2
S0 = Stats Data Bytes 1:0

Global 2 Register Names:

IS = Interrupt Source
IM = Interrupt Mask
M2 = MGMT Enables 2x
M0 = MGMT Enables 0x
FC = Flow Control Delays
M = Management

DM = Device Mapping
TK = Trunk Mask
TM = Trunk Mapping
IC = Ingress Rate Command
ID = Ingress Rate Data

CA = Cross Chip Port VLAN Addr
CD = Cross Chip Port VLAN Data
SM = Switch MAC
AS = ATU Stats
PO = Priority Overrides

SC = SMI PHY Command
SD = SMI PHY Data
WD = Watch Dog Control
QW = QoS Weights
SP = SDET Polarity

13.4 Switch Port Registers

Each Ethernet port in the devices contain their own per port registers. Each per port register is 16-bits wide and their bit assignments are shown in Figure 73.

Figure 73: Per Port Register Bit Map

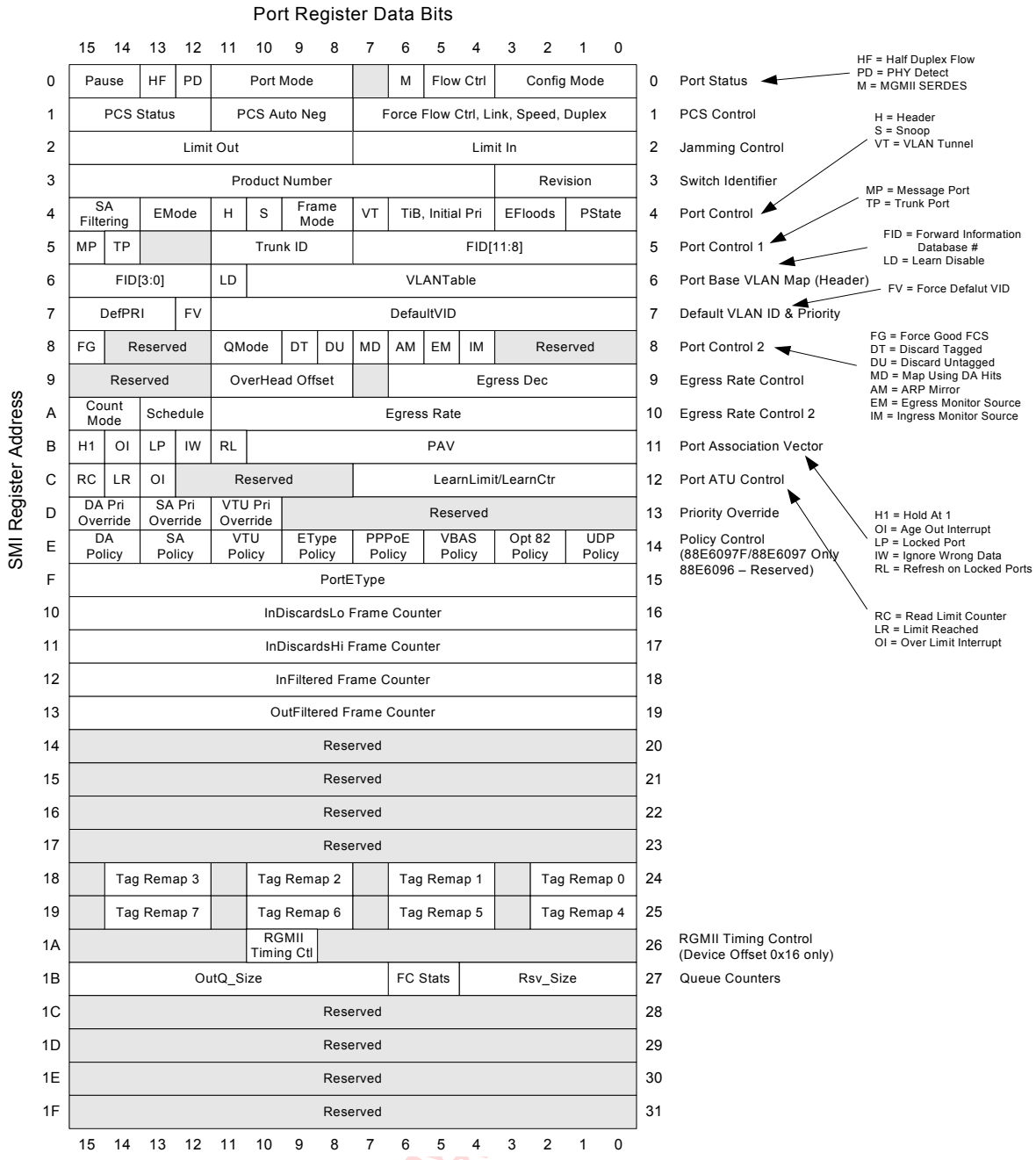


Table 71: Port Status Register[1]
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	PauseEn	RO	Pause Enabled bit. This indicates that full-duplex flow control will be used on this port if the port is in full-duplex mode. It is valid when the Link bit is a one. This bit reflects the Pause result from auto-negotiation only (i.e., the ForceFlowControl bit's value is not reflected in this bit). 0 = MAC Pause not implemented in the link partner or in MyPause 1 = MAC Pause is implemented in the link partner and in MyPause
14	MyPause	RWS or RWR	My Pause bit. This bit is sent to the PHY during PHY Polling Unit (PPU) initialization. This bit is not meaningful on ports that do not support Auto-Negotiation (i.e., internal ports). It is set high if FD_FLOW_DIS (Table 16) is low during RESETn. 0 = MAC Pause is not to be advertised as supported in this port 1 = MAC Pause is to be advertised as supported in this port
13	HdFlow	RO	Half-duplex Flow Control. This bit generally reflects the inverted value of the HD_FLOW_DIS bit during reset as follows: 0 = Half-duplex back pressure is not used on this port (unless the port's ForceFlowControl bit is set). 1 = Half-duplex back pressure is used on this port if this port is in a half-duplex mode.
12	PHYDetect	RWR	802.3 PHY Detected. This bits is set to a one if the PPU finds a non-all-one's value in either SMI registers 2 or 3 at the SMI device address 0x10 less than this address. 0 = An 802.3 PHY is not attached to this port 1 = An 802.3 PHY is attached to this port These bits are set at the end of the PPU's init routine (that is why this register must not be written to while the PPUState is Initializing – i.e., 01). The PPU poll routine uses these bits to determine which port's PHY to poll for Link, Duplex, Speed and Flow Control. If software changes these bits, the PPU changes its polling accordingly. A SWReset (Table 100) restarts the PPU's initialization and causes a re-write to this register (also in Table 100).
11	Link	RO	Link Status. This bit indicates the link status of the port as follows: 0 = Link is down 1 = Link is up The port's Link bit mirrors the LinkValue bit if the link is being forced by ForcedLink being a one (Table 73). Otherwise the port's Link bit takes the value of the source defined in Table 72.

Table 71: Port Status Register[1] (Continued)
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
10	Duplex	RO	Duplex mode. This bit is valid when the Link bit, above, is set to a one. 0 = Half-duplex 1 = Full-duplex The port's Duplex bit takes the value of the DpxValue bit if the duplex is being forced by ForcedDpx being a one (Table 73). Otherwise the port's Duplex bit takes the value of the source defined in Table 72.
9:8	Speed	RO	Speed mode. These bits are valid when the Link bit, above, is set to a one. When the port is configured in (G)MII mode the Speed bits are determined by the Px_MODE[2:0] bits. 00 = 10 Mbps 01 = 100 Mbps 10 = 1000 Mbps 11 = Reserved for future use The port's Speed bits take the value of ForceSpeed bits if the speed is being forced by ForceSpeed being non-0x3 (Table 73). Otherwise the port's Speed bits take the value of source defined in Table 72.
7	Reserved	RES	Reserved for future use.
6	MGMII (valid on Ports 8 to 10 only)	RWR or RWS[2]	SERDES Interface mode. When this bit is cleared to a zero and a PHY is detected connected to this port, the SERDES interface between this port and the PHY is SGMII. When this bit is set to a one and a PHY is connected to this port, the SERDES interface between this port and the PHY is MGMII. When this bit is set to a one and no PHY is detected on this port and the SERDES interface is being used, it is configured in 1000BASE-X mode. MGMII mode is recommended when connecting to Marvell® PHY devices that support auto-media detect when connecting to SERDES ports. MGMII mode uses out of band communication for link, speed, and flow control status, and works specifically with Marvell PHYs that support MGMII.
5	TxPaused	RO	Transmitter Paused. This bit is set to a one whenever the Rx MAC receives a Pause frame with a non-zero Pause time that is used by the Tx MAC (i.e., the transmitter is paused off). If the port is in half-duplex mode, this bit is never a one since all Rx Pause frames are ignored. If the port is in full-duplex mode this bit is never a one if Rx Pause frames are ignored because flow control is disabled on this port.
4	FlowCtrl	RO	Flow Control. This bit is set to a one whenever the Rx MAC determines that no more data should be entering this port. If the port is in half-duplex mode, this bit's being a one indicates that the port is using back pressure. If the port is in full-duplex mode this bit's being a one indicates that the port is going to or has sent a Pause frame with a non-zero Pause time to its link partner.

Table 71: Port Status Register[1] (Continued)
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
3	C_Duplex (valid on Port 9, Port 10 only)	RO	Config Duplex. This bit returns the port's duplex configuration mode determined at reset as follows: 0 = Half-duplex operation 1 = Full-duplex operation This bit is valid on port 9 and port 10 only and it reflects the value of P9_HALFDPX or P10_HALFDPX after reset regardless of the value of the P9_MODE[2:0] or P10_MODE[2:0] bits respectively after reset (if this bit is a one the port is not in half-duplex mode unless C_MODE = 01X).
2:0	C_Mode	RO	Config Mode. These bits return the port's interface type configuration mode determined by the value of the Px_MODE[2:0] pins at reset as follows: 000 = FD GMII with Px_GTXCLK = 125 MHz (1000BASE – Port 9 or Port 10 only) 001 = FD 1000BASE RGMII with P10_TXC = 125 MHz (Port 10 only) 010 = FD or HD MII with Px_GTXCLK = 25 MHz (100BASE – Port 9 or Port 10 only) 011 = FD or HD MII with Px_GTXCLK = 2.5 MHz (10BASE – Port9 or Port 10 only) 100 = FD cross-chip SERDES port 101 = FD 1000BASE-X Port (Port 8, Port 9, or Port 10 only) 110 = PHY Port (duplex and speed determined by the PPU) 111 = Port is disabled (Port 9 or Port 10 only) For Port 9 and Port 10 these bits are the port's Px_MODE[2:0] bits as read after reset. For Ports 0 to 7 these bits will be 111 if PHYDetect for this port is 0 or these bits will be 110 if PHYDetect for this port is 1. For Port 8 these bits will be 101 if P[8]_MODE is high, or these bits will be 100 if PHYDetect for this port is 0 and P[8]_MODE is low, or these bits will be 110 if PHYDetect for this port is 1 and P[8]_MODE is low.

1. This PortStatus register must not be written to if the PPUState in the Global Status register (Table 96) is 01.
2. The power on reset state of this bit on all ports is determined by the MGMII configuration pin.

The source of the port's Mode, Link, Speed and Duplex bits (assuming no forcing of the bits is occurring—Table 73) is defined in Table 72. Each of these bits, Link, Speed and Duplex, can be individually forced to any value by using the ForceXXX bits in Table 73.

Table 72: Port Configuration Matrix

Port #	Px_MODE[1]	PHY Detect[2]	C_Mode[3]	Port's Mode[4]	Link[5]	Speed[6]	Duplex[7]
0 to 7	No Pin	0	100	Disabled	0	100	FD
		1	110	Auto-Neg	PPU	PPU	PPU
8	0	0	100	Cross-chip	0	1000	FD
		1	110	AutoNeg	PPU	PPU	PPU
	1	X	101	1000BASE	PCS	1000	FD
9 or 10	0x0[8]	0	000	GMII	0	1000	FD
		1	000	GMII	PPU	PPU	PPU
	0x1	0	001	RGMII	0	1000	FD
		1	001[9]	Not Supported	-	-	-
	0x2	0	010	MII	0	100	P9_HALFDPX or P10_HALFDPX
		1	010	MII	PPU	PPU	PPU
	0x3	0	011	MII	0	10	P9_HALFDPX or P10_HALFDPX
		1	011	MII	PPU	PPU	PPU
	0x4	X	100	Cross-chip	0	1000	FD
	0x5	X	101	1000BASE	PCS	1000	FD
	0x6	X	110	Auto-Neg	PPU	PPU	PPU
	0x7	X	111	Disabled	0	1000	FD

1. Px_MODE is the single configuration pin for ports 8 or it is the three configuration pins for Port 9 and Port 10.
2. PHYDetect is the value of the PHYDetect bit for the port as seen in the port's Port Status register (Table 71).
3. C_Mode is the value that is seen in the C_Mode bits for this port (in the port's Port Status register - Table 71).
4. Port's Mode is the port's mode in name form as defined by the port's C_Mode bits.
5. Link is the source of the port's Link bit assuming Link is *not* being forced (Table 73). PPU means this bit comes from the PHY Polling Unit, PCS means this bit comes from the 1000BASE-X PCS block. Link bits that are '0' can be forced to a one by software (see the port's PCS Control register – Table 76). A port's Link bit as stored in the port's PPU Link register is cleared to zero whenever the port's PHYDetect transitions from a 1 to a 0. This ensures that the PPU's link for this port does not start out as a one, but transitions to a one only if the port's PHY still sees a link.
6. Speed is the source of the port's Speed bits assuming Speed is *not* being forced (Table 73). PPU means these bits comes from the PHY Polling Unit. 10, 100, or 1000 means the port defaults to this speed.
7. Duplex is the source of the port's Duplex bit assuming Duplex is *not* being forced (Table 73). PPU means this bit comes from the PHY Polling Unit. P9_HALFDPX or P10_HALFDPX means the bit comes from inverting the value seen on the P9_HALFDPX or P10_HALFDPX configuration pin respectively. FD means this mode defaults to full-duplex.
8. Px_MODE of 0x0 is NOT supported on Port 9 in the 176-pin LQFP package and must not be selected.
9. RGMII mode is supported on Port 10 only and only in full-duplex 1000BASE mode talking to a CPU. If an external PHY is connected in this mode data will not flow if the PHY links to 10 Mbps or 100 Mbps.

Table 73: PCS Control Register
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15[1]	PCSLink (Ports 8, 9 & 10 only)	RO	PCS Link up status. This bit b is a one whenever the PCS link is up.
14[1]	SyncOK (Ports 8, 9 & 10 only)	RO	This bit is set to a one when the PCS has detected a few comma patterns and is synchronized with its peer PCS layer.
13[1]	SyncFail (Ports 8, 9 & 10 only)	RO	This bit is set to a one if the PCS sees link. It will be cleared to a zero if the PCS has lost sync for 10 ms or more. Refer to IEEE 802.3 Clause 36.
12[1]	AnBypassed (Ports 8, 9 & 10 only)	RO	This bit indicates that Inband Auto-Negotiation was bypassed. If there is no reply during Auto-Negotiation, Bypass is activated and Link is set to 'up'. Speed is set to 1000 Mbps Speed and Duplex is set to Full-duplex. 1 = Auto-Negotiation is bypassed 0 = Auto-Negotiation is not bypassed
11[1]	AnBypass (Ports 8, 9 & 10 only)	RWR	In Band Auto-Negotiation Bypass Enable. This bit is relevant when Auto-Negotiation is done inband and not via the SMI interface (i.e., when InBandAn is set to a one). When this bit is set to a one and the link partner does not respond to the Auto-Negotiation process, the link is established by bypassing the Auto-Negotiation procedure (see AnBypassed bit). When this bit is cleared to a zero Auto-Negotiation cannot be bypassed.
10[1]	PCSAnEn (Ports 8, 9 & 10 only)	RWR or RWS	PCS Inband Auto-Negotiation Enable. PCS Inband Auto-Negotiation is done if the port is in 1000BASE-X for link and flow control support. This bit will be set to a one after reset mode if this port is in 1000BASE-X mode. When this port is in any other mode inband Auto-Negotiation is not done so this bit will be cleared to a zero after reset.
9[1]	RestartPCSAn (Ports 8, 9 & 10 only)	SC	Restart PCS Inband Auto-Negotiation. This bit is relevant when PCS Auto-Negotiation is done inband and not via the SMI interface (i.e., when InBandAn is set to a one). When this bit is set to a one the inband Auto-Negotiation restarts. This bit is reset to zero by the device immediately after restarting the Auto-Negotiation process.
8[1]	PCSAnDone (Ports 8, 9 & 10 only)	RO	This bit is cleared during the PSC Auto-Negotiation phase. It is set to one when PCS Auto-Negotiation is done (or if it were never done).
7	FCValue	RWR	Flow Control's Forced value. This bit is used to force flow control (if full-duplex) or backpressure (if half-duplex) to be enabled when the ForcedFC bit (below) is set to a one. Flow control/back pressure is forced enabled when this bit is set to a one. It is forced disabled when this bit is cleared to a zero. If the ForcedFC bit (below) is cleared to a zero, this bit has no effect. If Ingress Pause limiting is enabled (port offset 0x03) then this bit will be cleared to a zero if the Pause limit was reached on this port – so do not write to this register while Ingress Pause Limiting is enabled on this port.

Table 73: PCS Control Register (Continued)
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
6	ForcedFC		Force Flow Control. When this bit is set to a one flow control (if full-duplex) or backpressure (if half-duplex) for this port is forced to the value in the FCValue register (above) regardless of what the normal flow control value would be. In this way, flow control/backpressure can be forced to be enabled or disabled. When this bit is cleared to a zero, normal flow control detection occurs. If Ingress Pause limiting is enabled (port offset 0x03) then this bit will be set to a one if the Pause limit was reached on the port – so do not write to this register while Ingress Pause Limiting is enabled on this port.
5	LinkValue	RWR	Link's Forced value. This bit is used to force the link up or down when the ForcedLink bit (below) is set to a one. The link will be forced up when this bit is set to a one. It will be forced down when this bit is cleared to a zero. If the ForcedLink bit (below) is cleared to a zero this bit has no effect.
4	ForcedLink	RWR	Force Link. When this bit is set to a one the link for this port is forced to the value in the LinkValue register (above) regardless of what the normal link's value would be. In this way, the link can be forced to be up or down. When this bit is cleared to a zero, normal link detection occurs.
3	DpxValue	RWR	Duplex's Forced value. This bit is used to force the link to full- or half-duplex mode, when the ForcedDpx bit (below) is set to a one. The link duplex is forced to full when this bit is set to a one. It will be forced to half when this bit is cleared to a zero (Do not try to force half-duplex mode in a 1000BASE link - it is not supported and results are unpredictable). If the ForcedDpx bit (below) is cleared to a zero this bit has no effect.
2	ForcedDpx	RWR	Force Duplex. When this bit is set to a one the duplex for this port will be forced to the value in the DpxValue register (above) regardless of what the normal duplex's value would be. In this way the duplex can be forced to be full or half. When this bit is cleared to a zero, normal duplex detection occurs. **NOTE:** Only change the port's duplex when its link is down.
1:0	ForceSpd	RWS to 0x3	Force Speed. These bits are used to force the speed on this port as follows: 00 = 10 Mbps 01 = 100 Mbps 10 = 1000 Mbps 11 = Speed is not forced. Normal speed detection occurs. **NOTE:** Only change the port's speed when its link is down.

1. Bits 15:8 are irrelevant for the (G)MII mode.

Warning

The duplex and speed on a port must not be changed unless the link on the port is down.

Table 74: Jamming Control Register
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15:8	LimitOut	RWS To 0x9F	Limit the number of continuous Pause refresh frames that can transmitted from this port – assuming each Pause refresh is for the maximum pause time of 65536 slot times. When full-duplex Flow Control is enabled on this port, these bits are used to limit the number of Pause refresh frames that can be generated from this port to keep this port's link partner from sending any data. Setting these bits to 0x00 will allow continuous Pause frame refreshes to egress this port as long as this port remains congested. Setting these bits to 0x01 will allow 1 Pause frame to egress from this port for each congestion situation. Setting these bits to 0x02 will allow up to 2 Pause frames to egress from this port for each congestion situation, etc. The upper 3-bits of this register are a 2n multiplier for the lower 5 bits. The maximum count is 27 * 31 (7 comes from the upper 3 bits while 31 comes from the lower 5 bits). This equals 128 * 31 or 3,968 maximum Pause times.

Table 74: Jamming Control Register (Continued)
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
7:0	LimitIn	RWR	Limit the number of continuous Pause refresh frames that can be received on this port (if full-duplex) or the number of 16 consecutive collisions (if half-duplex). When a port has flow control enabled, these bits can be used to limit how long this port can be Paused or Back Pressured off to prevent a port stall through jamming. When these bits are in the range of 0x01 to 0xFF, and a frame is ready to be transmitted out this port, but it can't be transmitted due to the port being jammed, this limit mechanism starts. The limit mechanism starts counting new Pause refresh frames (Pause frames that re-load the Pause timer to other than 0x0000) or counts of 16 consecutive collisions. If the counter reaches the value contained in this register, the Port's ForceFC bit will be set to a one and its FCValue bit will be cleared to a zero and the global JamLimit Interrupt (Global 2, offset 0x00) will be set. This effectively disables Flow Control on the port once the Pause timer expires. If a frame gets transmitted out this port before the counter reaches this limit (i.e., the frame that was ready to be transmitted that started this process gets transmitted) then this limit mechanism counter resets back to zero. If the port is in half-duplex mode and Flow Control is disabled on the port, the JamLimit Interrupt will still be generated if the limit is reached on a frame. If Discard Excessive is set to a one (Global 1, offset 0x04) then the JamLimit Interrupted will never occur on half-duplex ports (since the frame is discarded after 16 consecutive collisions). Setting these bits to 0x00 will allow continuous jamming to be received on this port without the Port's ForceFC and FCValue bits getting modified. The modification of this Port's ForceFC and FCValue bits is the only indication that the limit was reached on this port.

Table 75: Switch Identifier Register
Offset: 0x03 or decimal 3

Bits	Field	Type	Description
15:4	Product Num	RO	Product Number or identifier: 88E6096 = 0x098 88E6097 and 88E6097F = 0x099
3:0	Rev	RO	Revision Identifier. The initial version of the devices has a Rev of 0x0. This Rev field may change at any time. Contact Marvell® FAEs for current information on the device revision identifier.

Table 76: Port Control Register
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
15:14	SA Filtering	RWR	Source Address Filtering controls. These bits select the SA (Source Address) filtering method to be used on the port as follows: 00 = SA Filtering Disabled – no frame will be filtered (i.e., discarded) due to the contents of its Source Address field 01 = Drop On Lock. Ingressing frames will be discarded if their SA field is not in the ATU's address database (i.e., its a new or unknown Source Address) or if this port's bit is not set in the PortVec bits for the frame's SA (i.e., this port is not the source port for that MAC address). Used for MAC based 802.1X. 10 = Drop On Unlock. Ingressing frames will be discarded if their SA field is in the ATU's address database as a Static entry with a PortVec of all zeros. Used to discard frames from known untrusted sources. 11 = Drop to CPU. Ingressing frames will be mapped to the CPUDest (global offset 0x1A) if their SA field is in the ATU's address database as a Static entry with a PortVec of all zeros and the frame is not otherwise filtered. Otherwise, the frames will be discarded if their SA field is not in the ATU's address database (i.e., its a new or unknown Source Address) or if this port's bit is not set in the PortVec bits for the frame's SA (i.e., this port is not the source port for that MAC address). This mode is a form of MAC based 802.1X where some frames can be forced to the CPU for further authentication prior to full authorization. SAFiltering[0] needs to zero when hardware address learn limiting is enabled on the port (Port ATU Control, offset 0x0C). In other words, hardware address learn limiting will work with either SA Filtering Disabled, or with SA Filtering set to Drop on Unlock.

Table 76: Port Control Register (Continued)
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
13:12	Egress Mode	RWR	Egress Mode. These bits determine how frames look when they egress this port. The effect of these bits is controlled by the Frame Mode bits below (bits 9:8 of this register) as follows: When Frame Mode = (00) Normal Network Frames these bits define the default tagging mode of the egressing frames. The default mode is used when the VID assigned to the frame during ingress is not contained in the VTU. The default modes are: 00 = Default to Unmodified mode – frames are transmitted unmodified[1] 01 = Default to Transmit all frames Untagged – remove the tag from any tagged frame 10 = Default to Transmit all frames Tagged – add a tag to any untagged frame 11 = Reserved for future use. When Frame Mode = (01) DSA Tag Frames these bits must remain at 00 as all other modes are 'Reserved for future use'. When Frame Mode = (10) Provider Tag Frames these bits must remain at 00 as all other modes are 'Reserved for future use'. When Frame Mode = (11) Ether Type DSA Tag Frames these bits define which frames get Ether Type DSA Tagged. In this case all Control frames egress with an Ether Type DSA Tag regardless of the setting of these bits (Control frames are To_CPU, From_CPU and To_Sniffer). Non-Control frames (i.e., Forward DSA Tag frames) will egress Ether Type DSA Tagged if these bits are 0b11, otherwise Forward frames will egress as Normal Network Frames as follows: 00 = Egress Forward DSA frames as Unmodified Normal Network Frames 01 = Egress Forward DSA frames as Untagged Normal Network Frames 10 = Egress Forward DSA frames as Tagged Normal Network Frames 11 = Egress all frames from this port with an Ether Type DSA Tag
11	Header	RWR	Ingress & Egress Header Mode. When this bit is set to a one all frames egressing this port are pre-pended with the Marvell® 2-byte Egress Header just before the frame's DA field. Also, all frames ingressing this port are expected to be pre pended with the Mavell 2-byte Ingress Header just before the frame's DA field. On Ingress the 1st 2 bytes after the SFD are removed from the frame and the frame's CRC and size is recomputed. If the frame's Ingress Header is non-zero it is used to update the port's VLAN Map register value (at register offset 0x06). When this bit is cleared to a zero, normal Ethernet frames egress the switch and are expected to ingress the switch. Header mode is intended to be used only on a port that is directly connected to a CPU that is performing routing as the Layer 3 portion of the frame becomes 32-bit aligned in the CPU's memory. It can be used in conjunction with DSA or Ether Type DSA Frame Modes (see Frame Mode bits below).

Table 76: Port Control Register (Continued)
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
10	IGMP/MLD Snoop	RWR	IGMP and MLD Snooping. When this bit is set to a one and this port receives an IPv4 IGMP frame or an IPv6 MLD frame, the frame is switched to the CPU port[2] overriding the destination ports determined by the DA mapping[3]. When this bit is cleared to a zero IGMP/MLD frames are not treated specially. IGMP/MLD Snooping is intended to be used on Normal Network or Provider ports (see Frame Mode bits below).
9:8	Frame Mode	RWR	Frame Mode. These bits are used to define the expected Ingress and the generated Egress tagging frame format for this port as follows: 00 = Normal Network 01 = DSA (Distributed Switch Architecture) 10 = Provider 11 = Ether Type DSA 00 = Normal Network mode uses industry standard IEEE 802.3ac Tagged or Untagged frames. Tagged frames use an Ether Type of 0x8100. Ports that are expected to be connected to standard Ethernet devices should use this mode. 01 = DSA mode uses a Marvell® defined tagged frame format for Chip-to-Chip and Chip-to-CPU connections. The extra data placed in the frame is needed to support the Spanning Tree Protocol (STP) as well as cross-chip features like Trunks, Mirrors, etc. Ports that are interconnected together to form a larger switch and ports connected to the management CPU must use this mode. 10 = Provider mode uses user definable Ether Types per port (see PortEType register, port offset 0x0F) to define that a frame is Provider Tagged. Ports that are connected to standard Provider network devices, or devices that use Tagged frames with an Ether Type other than 0x8100 should use this mode. Frames that ingress this port with an Ether Type that matches the port's PortEType will be considered tagged (for discarding policy), will have the tag's VID and PRI bits assigned to the frame (i.e., they will be used for switching and mapping), and will have the Provider Tag removed from the frame. If subsequent Provider Tags are found following the 1st Provider Tag, they too will be removed from the frame with their VID and PRI bits being ignored (if Remove1Tag is 0 in the Management register, Global (2), offset 0x05). Modified frames will be padded if required. Frames that ingress this port with an Ether Type that does not match the port's PortEType will be considered untagged (for discarding policy). The ingressing frames are modified so they are ready to egress out Customer ports (Normal Network Frame Mode ports) unmodified.

Table 76: Port Control Register (Continued)
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
9:8 (cont.)	Frame Mode (cont.)	RWR	Frames that egress this port will always have a tag added (even if they were already tagged). The added tag will contain this port's PortEType as its Ether Type. The PRI bits will be the Frame Priority (FPri) assigned to the frame during ingress. The VID bits will be the source port's Default VID bits (if the source port was in Normal Network mode), or the VID assigned to the frame during ingress (if the source port was in Provider mode or if the frame was DSA Tagged). 11 = Ether Type DSA mode uses standard Marvell® DSA Tagged frame information following a user definable Ether Type. This mode allows the mixture of Normal Network frames with DSA Tagged frames and is useful to be used on ports that connect to a CPU. Frames that ingress this port with an Ether Type that matches the port's PortEType will be considered DSA Tagged and processed accordingly. The frame's Ether Type and DSA pad bytes will be removed so the resulting frame will be ready to egress out Marvell DSA Tag Mode ports unmodified. Frames that ingress this port with a different Ether Type will be considered Normal Network Frames and processed accordingly. Marvell DSA Tag control frames (To_CPU, From_CPU and To_Sniffer) that egress this port will always get the port's PortEType inserted followed by two pad bytes of 0x00 before the DSA Tag. Marvell DSA Tag Forward frames that egress this port can egress just like the control frames (with the added Ether Type and pad) or they can egress as if the port was configured in Normal Network mode. This selection is controlled by the port's EgressMode bits above.
7	VLAN Tunnel	RWR	VLAN Tunnel. When this bit is cleared to a zero the port based VLANs defined in the VLANTable (Table 78), 802.1Q VLANs defined in the VTU (if 802.1Q is enabled - Table 76 and Table 101) and Trunk Masking (Table 137) are enforced for ALL frames. When this bit is set to a one the port based VLANTable masking, 802.1Q VLAN membership masking and the Trunk Masking is bypassed for any frame entering this port with a DA that is currently 'static' in the ATU. This applies to unicast as well as multicast frames.
6	TagIfBoth	RWS	Use Tag information for the initial QPri assignment if the frame is both tagged and its also IPv4 or IPv6 (and if InitialPri, below, = 0x3, Use Tag & IP Priority). The initial QPri is assigned as follows: 0 = QPri is frame's DiffServ bits (for IPv4) or the frame's Traffic Class bits (for IPv6) mapped using the IP PRI Mapping registers (Global 1 offsets 0x10 to 0x17). 1 = QPri is the determined FPri mapped using the IEEE PRI Mapping register (Global 1, offset 0x18).

Table 76: Port Control Register (Continued)
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
5:4	InitialPri	RWS to 0x3	Initial Priority assignment. Each frame entering a port is assigned a Frame Priority (FPri) and a Queue Priority (QPri). The FPri determined during ingress is written to the frame's IEEE 802.3ac tag PRI bits if the frame egresses tagged or if the frame egresses Provider Tagged (see EgressMode bits in – port offset 0x04). The QPri is used internally to determine which egress priority queue the frame is mapped into. On DSA ports (see FrameMode bits - port offset 0x04) the frame's default FPri is the DSA tag's PRI bits and the default QPri is FPri[2:1]. On non-DSA ports this register is used to select the frame's initial FPri & QPri depending upon the frame's type and content. These initial FPri & QPri assignments can be overridden by various overrides if enabled on the port (see port offset 0x0D, but FPri Overrides should not be enabled on DSA ports). If a frame does not meet the condition listed in the following table the defaults are assigned to frame. On non-DSA ports the default FPri is the port's DefFPri bit in the Default VLAN ID and Priority register at offset 0x07. The default QPri is obtained by mapping the frame's FPri value using the IEEE PRI Mapping register (Global 1, offset 0x18). The initial FPri and QPri on non-DSA ports are assigned as follows: 00 = Use Port defaults for FPri and QPri. 01 = Use Tag Priority. If the frame is tagged, FPri is set to the frame's tag PRI bits re-mapped by the port's Tag Remap registers (offset 0x17 & 0x18) and the QPri is the determined FPri mapped by the IEEE PRI Mapping register (Global 1, offset 0x18). If the frame is untagged the port defaults are used for FPri and QPri. 10 = Use IP Priority. If the frame is IPv4 or IPv6, QPri is the frame's DiffServ bits (for IPv4) or the frame's Traffic Class bits (for IPv6) mapped by the IP PRI Mapping registers (Global 1, offsets 0x10 to 0x17) and FPri[2:1] is the frame's QPri and FPri[0] is the port's DefFPri[0]. If the frame is not IPv4 nor IPv6 the port defaults are used for FPri and QPri, 11 = Use Tag & IP Priority. If the frame is tagged, FPri is the frame's tag PRI bits re-mapped by the port's Tag Remap registers (offset 0x17 & 0x18). If the frame is also IPv4 or IPv6 QPri's value will be determined by the TagIfBoth bit above. If the frame is untagged but it is IPv4 or IPv6, FPri and QPri are set according to the Use IP Priority setting above. If the frame is neither tagged nor IPv4 nor IPv6 the port defaults are used for FPri and QPri.

Table 76: Port Control Register (Continued)
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
3:2	Egress Floods	RWS to 0x3	Egress Flooding mode. The DA of every frame, unicast and multicast, is searched in the ATU. If the DA is found in the address database it is considered known. If it not found it is considered unknown. Frames with known DA's are not effected by this register. Frames with unknown DA's generally flood out all the ports (except for the port it originally came in on). This register can be used to prevent frames with unknown DA's from egressing this port as follows: 00 = Do not egress any frame with an unknown DA (unicast or multicast) 01 = Do not egress any frame with an unknown multicast DA 10 = Do not egress any frame with an unknown unicast DA 11 = Egress all frames with an unknown DA (unicast and multicast) The FloodBC (flood broadcast) bit in the Global 2 Management register (global 2, offset 0x05) is used to determine if Broadcast frames are considered multicast frames in the above description.
1:0	PortState	RWR Or RWS to 0b11[4]	Port State. These bits are used to manage a port to determine what kind of frames, if any, are allowed to enter or leave a port for simple bridge loop detection or 802.1D Spanning Tree. The state of these bits can be changed at any time without disrupting frames currently in transit. The Port States are: 00 = Disabled. The switch port is disabled and it will not receive or transmit any frames. The QC returns any pre-allocated ingress queue buffers when the port is in this mode. 01 = Blocking/Listening. The switch examines all frames without learning any SAs, and discards all frames. 10 = Learning. The switch will examine all frames, learning all SA address (except those from MGMT[5] frames), and still discard all but MGMT frames. It will allow MGMT frames only to exit the port. 11 = Forwarding. The switch examines all frames, learning SAs from all good frames (except those from MGMT frames), and receives and transmits all frames as a normal switch.

1. If this port has 802.1Q disabled and Cross-chip Port Based VLANs are being used in the switch, this port's EgressMode must be Default to Normal mode (to ensure the frames egress the switch looking exactly how they entered the switch) or Always add a Tag (to ensure the frames egress the switch with an extra tag compared to how they entered the switch).
2. The CPU port is determined by the CPUDest bits in Monitor Control Register (Global Offset 0x1A).
3. IGMP/MLD frames that are ingress filtered will not be sent to the CPUDest port.
4. The PortState bits for all ports come up in the Forwarding state unless the SW_MODE[1:0] pins are set to 0b00, the CPU attached mode, in which case the ports come up Disabled.
5. MGMT (management) frames are the only kind of frame that can be tunneled through Blocked ports. A MGMT frame is any frame whose multicast DA address appears in the ATU Database with the MGMT Entry State.

Table 77: Port Control 1
Offset: Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	Message Port	RWR or RWS[1]	Message Port. When the Learn2All bit in the ATU is set to a one (Table 109) learning message frames will be generated. These frames will be sent out all ports whose Message Port is set to a one. If this feature is used it is recommended that all DSA Tag ports, except for the CPU's port, have their MessagePort bit set to a one. Ports that are not DSA Tag ports (i.e., normal Network ports) should not have their MessagePort bit set to a one.
14	Trunk Port	RWR	Trunk Port. When this bit is set to a one, this port is considered to be a member of a Trunk with the Trunk ID defined below. When this bit is set to a zero, the port is treated as an individual port.
13:12	Reserved	RES	Reserved for future use.
11:8	Trunk ID	RWR	Trunk ID. When the Trunk Port bit (above) is set to a one these bits define which trunk this port is to be associated with. All ports that are members of the same trunk must be assigned the same Trunk ID and each group of ports that form a trunk must be assigned unique Trunk IDs.
7:0	FID [11:4]	RWR	Port's Default Forwarding Information Database (FID) bits 11:4. This field can be used with non-overlapping VLANs to keep each VLAN's MAC address mapping database separate from the other VLANs. This allows the same MAC address to appear multiple times in the address database (at most one time per VLAN) with a different port mapping per entry. This field is overridden by the FID returned from a VTU hit and it should be zero if not used. It must be a unique number for each independent, non-overlapping, FID if used. The lower four bits of the port's default FID are contained in the Port Base VLAN Map register (Table 78).

1. The Message Port bit on Ports 8 and 9 will be set to a one if the P9_TXD[3]/SW_24P configuration pin is high at the rising edge of RESETn.

Table 78: Port Based VLAN Map
Offset: Offset: 0x06 or Decimal 6[1]

Bits	Field	Type	Description
15:12	FID[3:0]	RWR	Port's Default Forwarding Information Database (FID) bits 3:0. This field can be used with non-overlapping VLANs to keep each VLAN's MAC address mapping database separate from the other VLANs. This allows the same MAC address to appear multiple times in the address database (at most one time per VLAN) with a different port mapping per entry. This field is overridden by the FID returned from a VTU hit and it should be zero if not being used. It needs to be a unique number for each independent, non-overlapping, VLAN, if used. The upper 4 bits of the port's default FID are contained in the Port Control 1 register (Table 77).
11	Learn Disable	RWR	Learn Disable. When this bit is cleared to a zero automatic learning on this port is controlled by the port's PAV bits (in the Port Association Vector Register at offset 0x0B—Table 83). When this bit is set to a one automatic learning does not occur for this port. This bit performs the same function as clearing the port's PAV bits but this bit is accessible by the CPU's Ingress Header so the CPU can enable and disable learning on a frame by frame basis.
10:0	VLANTable	RWS to all ones except for this port's bit	Port based VLAN Table. The bits in this table are use to restrict which output ports this input port can send frames to. The VLANTable bits are used for all frames, except for MGMT frames[2], even if 802.1Q is enabled on this port or if ProtectedPort is enabled on this port. These bits restrict where a port can send frames to (unless a VLANTunnel frame is being received – Table 76). To send frames to Port 0, bit 0 of this register must be a one. To send frames to Port 1, bit 1 of this register must be a one, etc. After reset, all ports are accessible since all the other port number bits are set to a one. This Port's bit is zero after reset. This prevents frames leaving the port on which they arrived. This Port's bit can to be set to a one in the devices, which allows frames to be switched back to the port on which they arrived. In view of this fact, care should be taken in writing code to manipulate these bits. This register is reset to 0x7FE for Port0 (SMI Device Address 0x10), and it resets to 0x7FD for Port1 (Addr 0x11),to 0x7FB for Port2 (Addr 0x12), etc. However, if the SW_24P configuration pin(on P9_TXD[3]) is set to one(24 Port Mode) Port 8 and 9 come up configured where they cannot communicate with each other (i.e., this register = 0x4FF for both Ports 8 & 9).

1. The contents of this register can be modified on a frame by frame basis if the port's Header Mode is enabled.
NOTE: Only the lower four bits of the FID field can be modified by the Header. Software that controls the FID field by using the Marvell® Header needs to take this into account. The DefaultVIDs used for Cross-Chip Port Based VLANs must be unique from the VIDs used for the 802.1Q VLANs currently active in the switch. Port Based VLAN ports need to have their frame's egress unmodified or the internal VID will be added to the frame if it is set to egress tagged.
2. See Section 8.3.1.

Table 79: Default Port VLAN ID & Priority, Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:13	DefPri	RWR	Default Priority. The bits of this register are used as the default ingress priority to use when no other priority information is available (neither is the frame IEEE Tagged, nor is it an IPv4 nor an IPv6 frame—or the frame is a priority type that is currently disabled (see InitialPri, port offset 0x04) and no other priority overrides are active on this frame. The DefPri bits are re-mapped by the IEEE-PRI Register (offset 0x18—Table 123) prior to being used.
12	Force DefaultVID	RWR	Force to use Default VID. When 802.1Q is enabled on this port (port offset 0x08) and this bit is set to a one, all ingress frames with IEEE 802.3ac Tags have their VID ignored and the port's DefaultVID below is used and replaced into the frame instead (if the frame dose not egress unmodified). When this bit is cleared to a zero all IEEE 802.3ac Tagged frames with a non-zero VID use the frame's VID unmodified. When 802.1Q is disabled on this port, this bit has no effect.
11:0	DefaultVID	RWS to 0x001	Default VLAN Identifier. When 802.1Q is enabled on this port the DefaultVID field is used as the IEEE Tagged VID added to untagged or priority tagged frames during egress that ingressed from this port. It is also used as a tagged frame's VID if the frame's VID was 0x000 (i.e., it is a priority tagged frame) or if the port's Force DefaultVID bit (above) is set to a one. When 802.1Q is disabled on this port, the DefaultVID field is assigned to all frames entering the port (if they are tagged or untagged[1]). This assignment is used internal to the switch, so only that Cross-chip Provider ports can be supported.

1. Port Based VLAN ports (ports where 802.1Q is disabled, port offset 0x08) need to have their frame's egress unmodified or the internal VID will be added to the frame if it is set to egress tagged.

Table 80: Port Control 2 Register
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15	ForceFCS	RWR	Force good FCS in the frame. When this bit is cleared to a zero frames entering this port must have a good CRC or else they are discarded. When this bit is set to a one the last four bytes of frames received on this port are overwritten with a good CRC and the frames are accepted by the switch (assuming that the frame's length is good and it has a destination).
14:12	Reserved	RES	Reserved for future use.
11:10	802.1QMode	RWR	IEEE 802.1Q Mode for this port. These bits determine if802.1Q based VLANs are used along with port based VLANs for this Ingress port. It also determines the action to be taken if an 802.1Q VLAN Violation is detected. These bits work as follows: 00 = 802.1Q Disabled. Use Port Based VLANs only. The VLANTable bits and the Cross-chip Port VLAN Table (global 2, offsets 0x0B & 0x0C) determine which Egress ports this Ingress port is allowed to switch frames to for all frames[1] (i.e., the frame's VID is ignored for switching and it's VID is not altered in the frame, i.e., all frames are considered untagged even if they are IEEE tagged). The VID assigned to the frame is the port's DefaultVID (port offset 0x07) to be used if the frame egresses a Provider port (see Frame Mode, port offset 0x04). 01 = Fallback. Enable 802.1Q for this Ingress port. Do not discard Ingress Membership violations and use the VLANTable bits below if the frame's VID is not contained in the VTU (both errors are logged – Table 101). 10 = Check. Enable 802.1Q for this Ingress port. Do not discard Ingress Membership violation but discard the frame if its VID is not contained in the VTU (both errors are logged – Table 101). 11 = Secure. Enable 802.1Q for this Ingress port. Discard Ingress Membership violations and discard frames whose VID is not contained in the VTU (both errors are logged – Table 101).
9	Discard Tagged	RWR	Discard Tagged Frames. When this bit is set to a one all non-MGMT frames that are processed as tagged are discarded as they enter this switch port. Priority only tagged frames (with a VID of 0x000) are considered untagged. This feature works if 802.1Q is enabled on the port or not (802.1Q Mode bits above). If the port is configured in Provider Mode (Frame Mode in Port Control, port offset 0x04) and this bit is set to a one, frames that contain an Ether Type that matches the port's PortETType (port offset 0x0F) that have a non-zero VID will be discarded. Discard Tagged should not be set on DSA or Ether Type DSA ports (see Frame Mode, port offset 0x04).

Table 80: Port Control 2 Register (Continued)
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
8	Discard Untagged	RWR	Discard Untagged Frames. When this bit is set to a one all non-MGMT frames that are processed as untagged are discarded as they enter this switch port. Priority only tagged frames (with a VID of 0x000) are considered untagged. This feature works if 802.1Q is enabled on the port or not (802.1Q Mode bits above). If the port is configured in Provider Mode (Frame Mode in Port Control, port offset 0x04) and this bit is set to a one, frames that don't contain an Ether Type that matches the port's PortETType (port offset 0x0F) and frames that contain an Ether Type that matches the port's PortETType that have a zero VID will be discarded. Discard UnTagged should not be set on DSA or Ether Type DSA ports (see Frame Mode, port offset 0x04).
7	MapDA	RWS	Map using DA hits. When this bit is set to a one, normal switch operation occurs where a frame's DA is used to direct the frame out of the correct ports. When this bit is cleared to a zero the frame will be sent out of the ports defined by EgressFloods (port offset 0x04) even if the DA is found in the address database. **NOTE:** If a multicast or unicast frame's DA is contained in the ATU with a MGMT Entry State the frame will be mapped out the port(s) defined by the ATU entry (i.e., the setting of the MapDA bit is ignored for MGMT frames).
6	ARP Mirror	RWR	ARP Mirror enable. When this bit is set to a one non-filtered Tagged or Untagged Frames that ingress this port that have the Broadcast Destination Address with an Ethertype of 0x0806 are mirrored to the CPUDest port (global offset 0x1A). This mirroring takes place after the ingress mapping decisions to allow ARPs to get to a CPU that is otherwise isolated. When this bit is cleared to a zero no special ARP handling will occur. This bit must not be set on DSA links or extra mirroring will result. ARP Mirror should not be set on DSA or Ether Type DSA ports (see Frame Mode, port offset 0x04).
5	Egress Monitor Source	RWR	Egress Monitor Source Port. When this bit is cleared to a zero, normal network switching occurs. When this bit is set to a one any frame that egresses out this port will also sent to the EgressMonitorDest Port (Table 124). The 802.1Q mode and VTU entries on the Egress Monitor Destination Port must be set the same as they are on the Egress Monitor Source port so the frames egress with the same tagged or untagged information. Egress Monitor Source should not be set on DSA or Ether Type DSA ports unless the port is directly connected to a CPU port and the CPU's code is being debugged (see Frame Mode, port offset 0x04).

Table 80: Port Control 2 Register (Continued)
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
4	Ingress Monitor Source	RWR	Ingress Monitor Source Port. When this bit is cleared to a zero normal network switching occurs. When this bit is set to a one, any frame that ingresses this port is also sent to the IngressMonitorDest Port (Table 124). The frame is sent to the IngressMonitorDest Port even if it is discarded owing to switching policy (like VLAN membership, etc.) but the frame will not be forwarded if its contains an error (such as CRC, etc.). Ingress Monitor Source should not be set on DSA or Ether Type DSA ports unless the port is directly connected to a CPU port and the CPU's code is being debugged (see Frame Mode, port offset 0x04).
3:0	Reserved	RES	Reserved for future use.

1. The VLANTable is sufficient to define Port Based VLANs when only one device is being used in a system (i.e., the VLANTable works for in-chip port based VLANs). When multiple devices are used in a system the Cross-Chip Port VLAN table (global 2, offset 0x0B & 0x0C) is used for frames entering a DSA port (or Ether Type DSA port if the Forward frames are DSA tagged, see Frame Mode, port offset 0x04). Both of these tables can be used with 802.1Q enabled using the VTU for frame VID switching (i.e., both port based and Q based VLAN are supported at the same time, in-chip and cross-chip).

Table 81: Egress Rate Control
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:8	Frame Overhead	RWR	Egress Rate Frame Overhead adjustment. This field is used to adjust the number of bytes that need to be added to a frame's IFG on a per frame basis. This is to compensate for a protocol mismatch between the sending and the receiving stations. For example if the receiving station were to add more encapsulations to the frame for the nodes further down stream, this per frame adjustment would help reduce the congestion in the receiving station. The egress rate limiter multiplies the value programmed in this field by four for computing the frame byte offset adjustment value (i.e., the amount the IPG is increased for every frame). This adjustment, if enabled, is made to every egressing frame's IPG and it is made in addition to any other IPG adjustments due to other Egress Rate Control settings. The egress overhead adjustment can add the following number of byte times to each frame's IPG: 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56 and 60. Example: If FrameOverhead = 0xB the egress rate limiter would increase the IPG between every frame by an additional 44 bytes. **NOTE:** When the Count Mode (port offset 0x0A) is in Frame based egress rate shaping mode, these Frame Overhead bits must be 0x0.
7	Reserved	RES	Reserved for future use.
6:0	Egress Dec	RWS 0x01	Egress Rate Decrement value. These bits indicate the Egress rate counter decrement value. Note that the rate at which the egress rate counter gets updated is still determined by the EgressRate field. This field determines the amount of decrement for each egress rate counter decrement update. The power on reset value for this field is 0x001 i.e., for every decrement the counter gets decremented by a value of 1. The expected values to be programmed for this field are: For any rate between 64kbps and 1Mbps: EgressDec = desired rate/ 64kbps For any rate between 1 Mbps and 100 Mbps: EgressDec = desired rate/ 1 Mbps For any rate between 100 Mbps and 1Gbps: EgressDec = desired rate / 10 Mbps

Table 82: Egress Rate Control 2
Offset: 0x0A or Decimal 10

Bits	Field	Type	Description
15:14	Count Mode	RWS to 0x2	Egress rate limiting count mode. These bits are used to indicate which bytes in the transmitted frames are counted for egress rate limiting as follows: 00 = Frame based 01 = Count all Layer 1 bytes 10 = Count all Layer 2 bytes 11 = Count all Layer 3 bytes Frame based: The egress rate limiting is done based on frame count as opposed to the byte count of the packet. Layer 1 = Preamble (8 bytes) + Frame's DA to CRC + IFG (12 bytes) + Header (if enabled – port offset 0x04) Layer 2 = Frames's DA to CRC Layer 3 = Frames's DA to CRC – 18[1] – 4 (if the frame is tagged)[2] A frame is considered tagged if the egress frame going out onto the wire is tagged. **NOTE:** When Count Mode is Frame based the Frame Overhead bits (port offset 0x09) must be 0x0.
13:12	Schedule	RWR	Port's Scheduling mode. 00 = Use an weighted round robin queuing scheme (default it 8, 4, 2, 1) 01 = Use Strict for priority 3 and use weighted round robin for priorities 2, 1 and 0 (default is 4, 2, 1) 10 = Use Strict for priorities 3 and 2 and use weighted round robin for priorities 1 and 0 (default is 2, 1) 11 = Use a Strict priority scheme for all priorities

Table 82: Egress Rate Control 2
Offset: 0x0A or Decimal 10

Bits	Field	Type	Description
11:0	Egress Rate	RWR	Egress data rate shaping. The EgressRate bits modify this port's effective transmission rate together with the EgressDec bits (Egress Rate Control, offset 0x09) and the CountMode bits (above). When this register is cleared to zero egress rate limiting is disabled. CountMode NOT Equal to 0x0 (Layer 1, 2 or 3 bytes): The devices use the following formula to limit the Egress data rate when the CountMode is NOT equal to 0x0: EgressRate = 8 bits * Egress Dec / (32 ns * Desired Egress Rate bits/sec) For example: CountMode = 0x2; Desired Rate = 640kbps; EgressDec = 640 kbps / 64 kbps = 0x00A EgressRate = 8 bits * 10 / (32ns * 640000 bits/sec) = 3906 or 0xF42 If CountMode is not equal to zero, the desired rate can vary from 64 kbps to 1 Gbps in the following increments: Desired rate between 64 kbps and 1 Mbps in increments of 64 kbps. Desired rate between 1 Mbps to 100 Mbps in increments of 1 Mbps. Desired rate between 100 Mbps to 1 Gbps in increments of 10 Mbps. CountMode Equal to 0x0 (Frame rate): The devices use the following formula to limit the Egress data rate when the CountMode is equal to 0x0: EgressRate = EgressDec / (32 ns * Desired Egress Rate frames/sec) Where EgressDec is recommended to be programmed to a 0x01 when CountMode = 0x0. For example: CountMode = 0x0; Desired Rate = 10k frames per second Frame size is assumed to be 64Bytes and EgressDec is assumed to be 0x1. EgressRate = 1 / (32ns * 10000 frames/sec) = 3125 or 0xC34 In CountMode = 0x0, the desired frame rate can vary from 7.6k to 1.488M frames per second. Egress Rate Shaping transmits a frame at wire speed counting the transmitted bytes determined by CountMode above. The value in this register determines the time it takes for the transmitted byte count to reach zero. When it reaches zero, the next frame is allowed to be transmitted and the process repeats. This burstless rate shaping is the best method for supporting the minimal amount of buffering required in the link partner this device is connected to.

1. The 18 bytes are: 6 for DA, 6 for SA, 2 for EtherType and 4 for CRC.
2. Only one tag is counted even if the frame contains more that one tag (i.e. it is Provider Tagged).

Table 83: Port Association Vector
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
15	HoldAt1	RWR	Hold Aging ATU Entries at an Entry State value of 1. When this bit is cleared to a zero normal Aging occurs for ATU entries associated with this port. When this bit is set to a one ATU entries associated with this port (either directly or indirectly because the entry contained a Trunk association) will age down to an Entry State of 0x1 but will not go to 0x0 (0x0 would purge the entry).
14	IntOn AgeOut	RWR	Interrupt on Age Out. When aging is enabled, all address entries in the ATU's address database are periodically aged (non-static entries have their EntryState bits decremented by 1 until they reach the value of 0x01 or 0x0). When a non-static entry is aged from an EntryState value of 0x1, and if that entry is associated with this port (either directly or indirectly because the entry contained a Trunk association), and if this IntOnAgeOut bit is set to a one, an AgeOutViolation (global 0x0B) will be captured for that entry. If the port's HoldAt1 bit (above) is zero then ATU entries will automatically age out (i.e., their EntryState will be written back to 0x0). But if the port 's HoldAt1 bit is one aging entries with an EntryState = 0x1 will remain with not be automatically purged (i.e., their EntryState will remain at 0x1).
13	LockedPort	RWR	Locked Port. When this bit is cleared to a zero, normal address learning will occur. When this bit is set to a one CPU directed learning (needed for 802.1X MAC authentication) is enabled on this port. In this mode, an ATU Miss Violation interrupt occurs when a new SA address is received in a frame on this port. Automatically SA learning and refreshing is disabled in this mode. If the ATUAgeIntEn (global 2, offset 0x05) is enabled then ATU Miss Violations will also occur if a frame's SA is already in the address database, but it has an EntryState less than 0x4 (i.e., the entry is about half way aged out). Station moves will not auto refresh and will generate an ATU Miss Violation. This is done so CPU directed learning can refresh entries still being used before they age out. If RefreshLocked (below) is enabled then auto refreshing of known addresses will occur even if this port is Locked. No ATU Miss Violations from known addresses will occur either (regardless of the setting of the ATUAgeIntEn bit). This bit needs to be cleared to a zero when hardware address learn limiting is enabled on the port (Port ATU Control, offset 0x0C) so auto learning will occur before the limit is reached.

Table 83: Port Association Vector
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
12	Ignore WrongData	RWR	Ignore Wrong Data. All frame's SA addresses are searched for in the ATU's address database. If the frame's SA address is found in the database and if the entry is 'static' (see Section 8.8.1) or if the port is 'locked' (see bit 13 above), the source port's bit is checked to ensure the SA has been assigned to this port. If the SA is NOT assigned to this port it is considered an ATU Member Violation. If the IgnoreWrongData bit is cleared to a zero, an ATU Member Violation interrupt will be generated. If the IgnoreWrongData bit is set to a one the ATU Member Violation error is masked and ignored.
11	Refresh Locked	RWR	Auto Refresh known addressed when port is Locked. Already known addresses will be auto refreshed (i.e., their Entry State will be updated to 0x7 whenever this address is used as a source address in a frame on this port) even when this port is Locked (see the LockedPort bit above) when this bit is set to a one. Station moves are not auto refreshed in this mode (i.e., the normal station move interrupt is generated if IgnoreWrongData, bit 12 above, is cleared). When this bit is cleared to a zero auto refreshing will not occur on Locked ports.
10:0	PAV	RWS to all zeros except for this port's bit	Port Association Vector for ATU learning. The value in these bits is used as the port's DPV on automatic ATU Learning or Entry_State refresh whenever these bits contain a non-zero value. When these bits are all zero, automatic Learning and Entry_State refresh is disabled on this port. For normal switch operation, this port's bit should be the only bit set in the vector. These bits must only be changed when frames are not entering the port (see PortState bits in Port Control – Table 76). The PAV bits can be used to set up port trunking (along with the VLANTable bits ()). For the two ports that form a trunk, set both of their port's bits in both port's PAV registers, then use the VLANTable (port offset 0x06) to isolate the two ports from each other, or to use the Trunk Mask table (Global 2 offset 0x07) to steer the traffic from the other ports down the desired trunk line of the pair using DA/SA Load Balancing.

Table 84: Port ATU Control[1]
Offset: 0x0c or Decimal 12

Bits	Field	Type	Description
15	Read LearnCnt	RWR	Read the current number of 'active' unicast MAC addresses associated with this port. When this bit is cleared to a zero the LearnLimit bits are accessible below. When this bit is set to a one the LearnCnt bits are accessible below. Note: Writing this bit to a 0 will cause the bits 7:0 to be written to the LearnLimit register (i.e., a single write to this register that results with this bit being 0 will also write bit 7:0 to the LearnLimit register). A non-destructive 1 to 0 transition of this bit can be accomplished as long as bits 7:0 are written to the same value that they were prior to setting this bit to a 1[2].
14	Limit Reached	RO	Limit Reached. This bit is set to a one when the port can no longer auto learn any more MAC addresses because the address learn limit set on this port has been reached. When this bit is set to a one the device will act as if the port is Locked (port offset 0x0B) and the SA Filtering mode is Drop on Lock (port offset 0x04). The port's LockedPort and SAFiltering bits will not change in value, however. In fact the LockedPort and SAFiltering[0] bits must be, and stay zeros for the hardware address learn limiting to work properly. SAFiltering[1] can be zero or one allowing address learn limiting to work with the Drop On UnLock mode.
13	OverLimit IntEn	RWR	Over Limit Interrupt Enable. An ATU Miss Violation will be generated when this bit is set to a one and a new source address is trying to be auto learned, but can't, because the Limit Reached bit, above, is set. Clearing this bit to a zero will prevent ATU Miss Violations in this case.

Table 84: Port ATU Control[1]
Offset: 0x0c or Decimal 12

Bits	Field	Type	Description
12:8	Reserved	RES	Reserved for future use.
7:0	LearnLimit/ LearnCnt	RWR/RO	Port's Auto Learning Limit or port's current Auto Learning count. When the ReadLearnCnt bit above is cleared to zero these bits are used to enable Auto Learning limits on the port as defined below. In this mode the reading and writing of this register goes to the LearnLimit register. When the ReadLearnCnt bit is set to a one these bits are used to read back the port's current Auto Learning counter (LearnCnt). In this mode writing to these bits will have no effect (so read/modify/write operations to the ReadLearnCnt bit to toggle modes can still be done). When ReadLearnCnt = 0 and these bits are cleared to zero, normal address learning and frame policy occurs. When ReadLearnCnt = 0 and these bits are set to a non-zero value and the port is not a member of a Trunk (port offset 0x05), the number of MAC addresses that can be learned on this port are limited to the number defined in these bits. Automatic learning and frame policy will occur normally until the number of unicast MAC addresses auto-learned from this port reaches this port's LearnLimit (addresses that were learned from this port but were aged out are not counted – i.e., this register limits the number of 'active' unicast MAC addresses associated to this port). When the LearnLimit has been reached any frame that ingresses this port with a source MAC address not already in the address database that is associated with this port will be discarded (the port will act as if the port is Locked and the port's DropOnLock SAFIltering mode is set). Normal auto-learning will resume on the port as soon as the number of 'active' unicast MAC addresses associated to this port is less than the LearnLimit (due to address aging). When ReadLearnCnt = 1 these bits become read only and return the current number of 'active' unicast MAC addresses associated to this port. Note: The LearnCnt counter will be held at zero if the LearnLimit = 0 (i.e., whenever the limit function is disabled the LearnCnt is re-initialized). This feature will not work when this port is configured as a Trunk port (port offset 0x05). The only CPU directed ATU Operations that effect the LearnCnt counter is the ATU Flush All Entries and the ATU Flush All Non-Static Entries. In both cases the LearnCnt is cleared to zero. This means that a CPU directed ATU Load, Purge or Move of one or more unicast addresses associated with this port will not have any effect on the LearnCnt's value. Care is needed when enabling this feature. 1st disable learning on the ports. 2nd flush all non-static addresses in the ATU. 3rd define the desired limit for the ports. 4th re-enable learning on the ports.

1. This hardware Learn Limit feature requires Learn2All must = 1 (global offset 0x0A).
2. If the LearnLimit is set to a value that is different from what it was before reading the LearnCtr unpredictable results will occur. It is best to set the LearnLimit prior to taking the port out of the Disabled or Blocking Port State (port offset 0x04). If the LearnLimit must be changed, Block the port, clear the LearnCtr (set LearnLimit to 0x00) and Move all non-Static ATU entries from this port to port 0xF(to disassociate all entries from this port - global offset 0x0B) prior to setting the new LearnLimit's value.

Table 85: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
15:14	DAPri Override	RWR	DA Priority Override. When these bits are cleared to a zero normal frame priority processing occurs. When either of these bits are set to a one then DA ATU priority overrides can occur on this port. A DA ATU priority override occurs when the destination address of a frame results in an ATU hit where the DA's MAC address returns an EntryState that indicates Priority Override. When this occurs three forms of priority overrides are possible: If DAPriOverride[0] is set to a one, PRI value assigned to the frame's DA (in the ATU database) is used to overwrite the frame's previously determined FPri (frame priority). If the frame egresses tagged the priority in the frame will be this new PRI value. DA frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, port offset 0x04). If DAPriOverride[1] is set to a one, the two upper bits of the PRI value assigned to the frame's DA (in the ATU database) is used to overwrite the frame's previously determined QPri (queue priority). The QPri is used internally to map the frame to one of the egress queues inside the switch. QPri override will not affect the contents of the frame in any way. DA queue priority override needs to be set on DSA ports to keep the frame in the correct queue Cross-chip. If both DAPriOverride bits are a one then both the above overrides take place on the frame. The DA ATU Priority Override has highest priority over the port's Default Priority, the frame's IEEE and/or IP priorities, the VTU Priority Override and the SA Priority Override. **NOTE:** If a frame's DA is contained in the ATU with a MGMT Entry State the frame's priority will be overridden regardless of the state of this bit.

Table 85: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
13:12	SAPri Override	RWR	SA Priority Override. When these bits are cleared to a zero normal frame priority processing occurs. When either of these bits are set to a one then SA ATU priority overrides can occur on this port. An SA ATU priority override occurs when the source address of a frame results in an ATU hit where the SA's MAC address returns an EntryState that indicates Priority Override. When this occurs three forms of priority overrides are possible: If SAPriOverride[0] is set to a one, PRI value assigned to the frame's SA (in the ATU database) is used to overwrite the frame's previously determined FPri (frame priority). If the frame egresses tagged the priority in the frame will be this new PRI value. SA frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, port offset 0x04). If SAPriOverride[1] is set to a one, the two upper bits of the PRI value assigned to the frame's SA (in the ATU database) is used to overwrite the frame's previously determined QPri (queue priority). The QPri is used internally to map the frame to one of the egress queues inside the switch. QPri override will not affect the contents of the frame in any way. SA queue priority override needs to be set on DSA ports to keep the frame in the correct queue Cross-chip. If both SAPriOverride bits are a one then both the above overrides take place on the frame. The SA ATU Priority Override has higher priority than the port's Default Priority, the frame's IEEE and/or IP priorities, and the VTU Priority Override. The priority determined by the frame's SA can be overridden, however, by the frame's DA Priority Override.

Table 85: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
11:10	VTUPri Override	RWR	VTU Priority Override. When these bits are cleared to a zero normal frame priority processing occurs. When either of these bits are set to a one then VTU priority overrides can occur on this port. A VTU priority override occurs when the determined VID of a frame[1] results in a VID whose VIDPRIOverride bit in the VLAN database is set to a one. When this occurs three forms of priority overrides are possible: If VTUPriOverride[0] is set to a one, the VIDPRI value assigned to the frame's VID (in the VLAN database) is used to overwrite the frame's previously determined FPri (frame priority). If the frame egresses tagged the priority in the frame will be this new VIDPRI value. VID frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, port offset 0x04). If VTUPriOverride[1] is set to a one, the VIDPRI value assigned to the frame's VID (in the VLAN database) is used to overwrite the frame's previously determined QPri (queue priority). The QPri is used internally to map the frame to one of the egress queues inside the switch. QPri override will not affect the contents of the frame in any way. VID queue priority override needs to be set on DSA ports to keep the frame in the correct queue cross-chip. If both VTUPriOverride bits are a one then both the above overrides take place on the frame. The VTU Priority Override has higher priority than the port's Default Priority and the frame's IEEE and/or IP priorities. The priority determined by the frame's VID can be overridden, however, by the frame's SA and/or DA Priority Overrides.
9:0	Reserved	RES	Reserved for future use.

1. The VID of a frame could be a tagged frame's VID or the port's DefaultVID.

Note

The Policy Control Register is only available for the 88E6097F and 88E6097 devices. These registers are reserved in the 88E6096 device.

Table 86: Policy Control Register[1]
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
15:14	DA Policy	RWR	DA Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then DA Policy Mapping can occur on this port. DA Policy Mapping occurs when the DA of a frame is contained in the ATU address database with an Entry State that indicates Policy (global offset 0x0C). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.
13:12	SA Policy	RWR	SA Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then SA Policy Mapping can occur on this port. SA Policy Mapping occurs when the SA of a frame is contained in the ATU address database with an Entry State that indicates Policy (global offset 0x0C). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.

Table 86: Policy Control Register[1] (Continued)
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
11:10	VTU Policy	RWR	VTU Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then VTU Policy Mapping can occur on this port. VTU Policy Mapping occurs when the VID of a frame[2] is contained in the VTU database with the VidPolicy bit set to a one (global offset 0x02). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.
9:8	EType Policy	RWR	EType Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then EType Policy Mapping can occur on this port if the port's FrameMode is Normal Network (port offset 0x04). EType Policy Mapping occurs when the EtherType of a frame matches the PortEType register (port offset 0x0F). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.

Table 86: Policy Control Register[1] (Continued)
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
7:6	PPPoE Policy	RWR	PPPoE Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then PPPoE Policy Mapping can occur on this port. PPPoE Policy Mapping occurs when the EtherType of a frame matches 0x8863. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1C) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1C) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.
5:4	VBAS Policy	RWR	VBAS Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then VBAS Policy Mapping can occur on this port. VBAS Policy Mapping occurs when the EtherType of a frame matches 0x8200. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.

Table 86: Policy Control Register[1] (Continued)
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
3:2	Opt82 Policy	RWR	DHCP Option 82 Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then DHCP Option 82 Policy Mapping can occur on this port. DHCP Option 82 Policy Mapping occurs when the ingressing frame is an IPv4 UDP with a UDP Destination port = 0x0043 (or decimal 67) or 0x0044 (or decimal 68) or an IPv6 UDP with a UDP Destination port = 0x0223 or 0x0222. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.
1:0	UDP Policy	RWR	UDP Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then UDP Policy Mapping can occur on this port. UDP Policy Mapping occurs when the ingressing frame is a Broadcast IPv4 UDP or a Multicast IPv6 UDP. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUpDest port (global offset 0x1A) 11 = Discard (filter) the frame Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.

1. Policy should only be performed on Normal or Provider ports (see Frame Mode, port offset 0x04).
2. The VID of a frame could be a tagged frame's VID or the port's DefaultVID.

Table 87: Port E Type
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
15:0	Port EType	RWS to 0x9100	Port's Special Ether Type. This Ether Type is used for many features depending upon the mode of the port (as defined by the port's EgressMode and FrameMode bits – in Port Control, port offset 0x04). If the port's FrameMode is Normal Network mode, this register's value can be used to Trap, Mirror or Discard frames that ingress this port with this Ether Type (see ETypePolicy register at port offset 0x0D). If the port's FrameMode is Provider mode, this register's value is used as the Provider Tag Ether type added to frames that egress this port. It is also used as the expected Provider Tag Ether type on frames that ingress this port. The removal of the Provider Tags during ingress 'normalizes' the frame in memory so it can be switched to Customer ports or to another Provider Port (where it will get that port's PortEType added as the Provider Tag Ether type). If the port's FrameMode is Ether type Marvell® DSA Tag mode, this register's value is used as the Marvell DSA Ether type added to the appropriate frames that egress this port (either all frames on just control frames as determined by the port's EgressMode bit, offset 0x04). It is also used to match an ingressing frame's Ether type to indicate which frames contain a Marvell DSA Ether type tag. Frames that contain an Ether typed Marvell DSA Tag are 'normalized' during ingress to be stored in memory as non-Ether typed Marvell DSA tagged frames.

Table 88: InDiscards Low Counter
Offset: 0x10 or Decimal 16

Bits	Field	Type	Description
15:0	InDiscardsLo	RO	InDiscards Low Frame Counter. This counter contains the lower 16-bits of the 32-bit InDiscards counter. This 32-bit counter increments each time a good, non-filtered, frame is received but it cannot be forwarded owing to a lack of buffer memory. The counter wraps back to zero. The only time this counter does not increment is when this port is Disabled (see PortState Table 76). This register will be cleared by a Flush All Counters for this port or all ports StatsOp command (Table 127).

Table 89: InDiscards High Counter
Offset: 0x11 or Decimal 17

Bits	Field	Type	Description
15:0	InDiscardsHi	RO	InDiscards High Frame Counter. This counter contains the upper 16-bits of the 32-bit InDiscards counter. This 32-bit counter increments each time a good, non-filtered, frame is received but it cannot be forwarded due to a lack of buffer memory. The counter wraps back to zero. The only time this counter does not increment is when this port is Disabled (see PortState Table 76). This register is cleared by a Flush All Counters for this port or all ports StatsOp command (Table 127).

Table 90: InFiltered Counter
Offset: 0x12 or Decimal 18

Bits	Field	Type	Description
15:0	InFiltered	RO	InFiltered Frame Counter. This 16-bit counter gets incremented each time a good frame enters this port that was not forwarded due to filtering rules. The counter wraps back to zero. The only time this counter does not increment is when this port is Disabled (see PortState Table 76). This register will be cleared by a Flush All Counters for this port or all ports StatsOp command (Table 127).

Table 91: OutFiltered Counter
Offset: 0x13 or Decimal 19

Bits	Field	Type	Description
15:0	OutFiltered	RO	OutFiltered Frame Counter. This 16-bit counter gets incremented each time a good frame enters this port that was not forwarded due to filtering rules. The counter wraps back to zero. The only time this counter does not increment is when this port is Disabled (see PortState Table 76). This register will be cleared by a Flush All Counters for this port or all ports StatsOp command (Table 127).

Table 92: Port IEEE Priority Remapping Registers
Offset: 0x18 or Decimal 24

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:12	TagRemap3	RWS to 0x3	Tag Remap 3. All IEEE tagged frames with a priority of 3 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
11	Reserved	RES	Reserved for future use.
10:8	TagRemap2	RWS to 0x2	Tag Remap 2. All IEEE tagged frames with a priority of 2 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
7	Reserved	RES	Reserved for future use.
6:4	TagRemap1	RWS to 0x1	Tag Remap 1. All IEEE tagged frames with a priority of 1 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
3	Reserved	RES	Reserved for future use.
2:0	TagRemap0	RWR	Tag Remap 0. All IEEE tagged frames with a priority of 0 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.

Table 93: Port IEEE Priority Remapping Registers
Offset: 0x19 or Decimal 25

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:12	TagRemap7	RWS to 0x7	Tag Remap 7. All IEEE tagged frames with a priority of 7 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
11	Reserved	RES	Reserved for future use.
10:8	TagRemap6	RWS to 0x6	Tag Remap 6. All IEEE tagged frames with a priority of 6 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
7	Reserved	RES	Reserved for future use.
6:4	TagRemap5	RWS to 0x5	Tag Remap 5. All IEEE tagged frames with a priority of 5 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
3	Reserved	RES	Reserved for future use.
2:0	TagRemap4	RWS to 0x4	Tag Remap 4. All IEEE tagged frames with a priority of 4 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.

Table 94: RGMII Timing Control (Device Offset 0x16 only)
Offset: 0x1A or Decimal 26

Bits	Field	Type	Description
15:11	Reserved	WBAR	Write back as read.
10	RGMII Receive Timing Control	RWR to 0	Changes to this bit are disruptive to the normal operation; hence, any change to this register must not be done while the port is linked. 0 = Default 1 = Add delay to RXC for RXD outputs See Section 16.9.1 for RGMII Timing modes.
9	RGMII Transmit Timing Control	RWR to 0	Changes to this bit are disruptive to the normal operation; hence, any change to this register must not be done while the port is linked. 0 = Default 1 = Add delay to TXC for TXD outputs See Section 16.9.1 for RGMII Timing modes.
8:0	Reserved	WBAR	Write back as read.

Note

These RGMII timing bits affect Port 10's timing only and only if Port 10 is configured in RGMII mode (Section 2.2).

Note

The location of these bits could change in future devices.

Table 95: Queue Counter Registers
Offset: 0x1B or Decimal 27

Bits	Field	Type	Description
15:7	OutQ_Size	RO	Egress Queue Size Counter. This counter reflects the current number of Egress buffers switched to this port. This is the total number of buffers across all four priority queues.
6	BufHigh	RO	Output from QC telling the MAC that it should perform Flow Control
5	Fc_En	RO	Input into the QC telling it that Flow Control is enabled on this port.
4:0	Rsv_Size	RO	Ingress Reserved Queue Size Counter. This counter reflects the current number of reserved Ingress buffers assigned to this port.

13.5 Switch Global Registers

The devices contain global registers that affect all Ethernet ports in the device. Each global register is 16-bits wide. Global registers' bit assignments are shown in Figure 74.

Figure 74: Global Register Bit Map

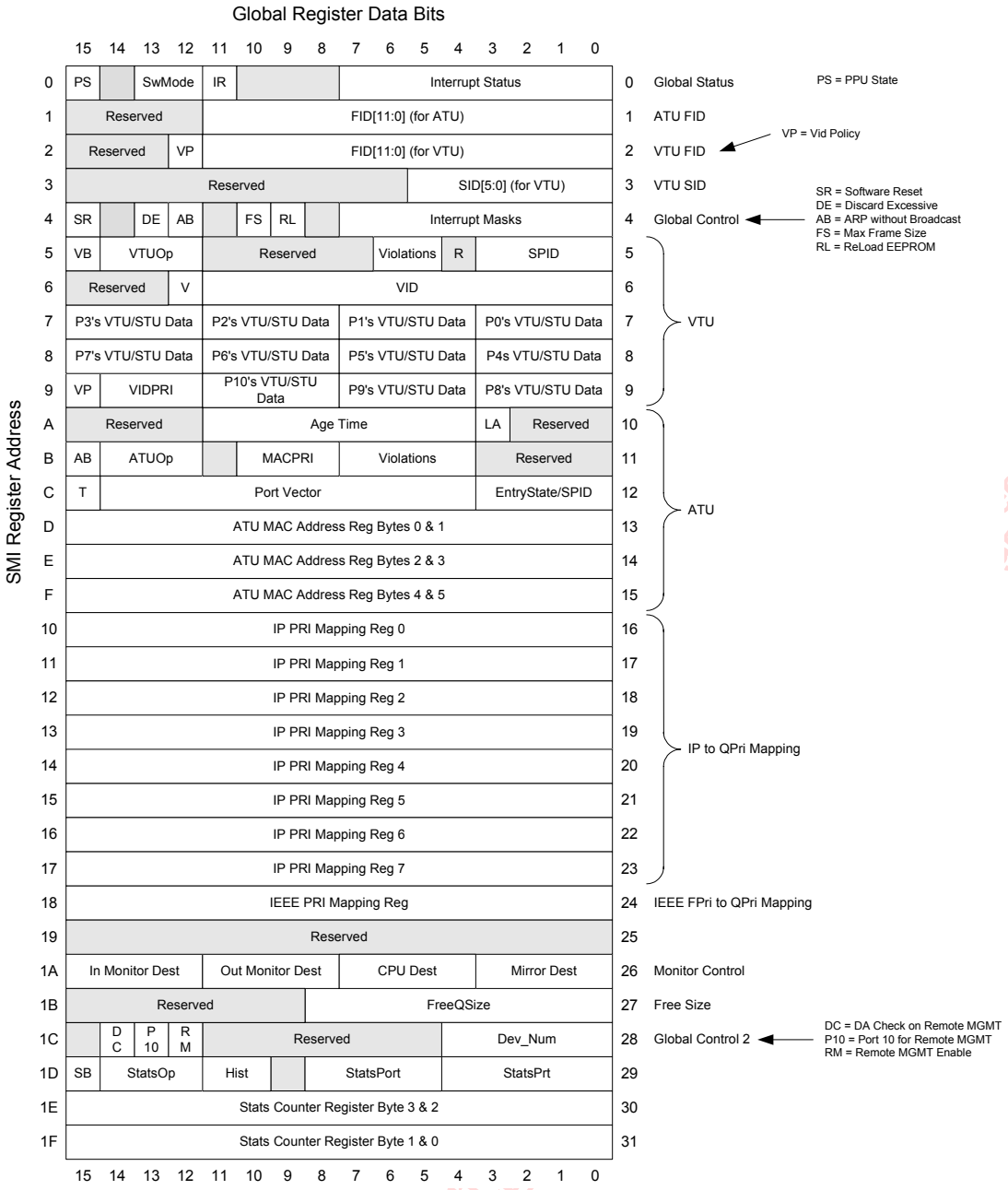


Table 96: Switch Global Status Register
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	PPUState	RO	PHY Polling Unit State. These bits indicate the state of the PPU as follows: 0 = PPU is Active detecting and initializing external PHYs. The PortStatus registers (Table 71) must not be written by software. 1 = PPU Polling. This indicates the PPU is Active polling the external PHYs. Software can write to the PortStatus registers (Table 71).
14	Reserved	RES	Reserved for future use.
13:12	SW_Mode	RO	Switch Mode. These bits return the value of the SW_MODE[1:0] pins.
11	InitReady	RO	SwitchReady. This bit is set to a one when the Address Translation Unit, the VLAN Translation Unit, the Queue Controller and the Statistics Controller complete their initialization and are ready to accept frames.
10:8	Reserved	RES	Reserved for future use.
7	DeviceInt	RO	Device Interrupt. This bit is set to a one when any of the device interrupts have at lease one active interrupt. The device interrupts are defined in the Interrupt Source register (Global 2, offset 0x00). This bit being high will cause the device's INTn pin to go low if the DevIntEn bit in Global Control (global offset 0x04) is set to a one.
6	StatsDone	LH	Statistics Done Interrupt. This bit is set to a one whenever the STATBusy bit (Table 127) transitions from a one to a zero. It is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the STATDoneIntEn bit in the Global Control register (Table 100) is set to a one.
5	VTUProb	RO	VLAN Table Problem/Violation Interrupt. This bit is set to a one if a VLAN Violation is detected. It is automatically cleared when all of the pending VTU Violations have been serviced by the VTU Get/Clear Violation Data operation (Table 101). This bit being high causes the device's INTn pin to go low if the VTUProbIntEn bit in Global Control (Table 100) is set to a one.
4	VTUDone	LH	VTU Done Interrupt. This bit is set to a one whenever the VTUBusy bit (Table 101) transitions from a one to a zero. It is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the VTUDoneIntEn bit in Global Control (Table 100) were set to a one.
3	ATUProb	RO	ATU Problem/Violation Interrupt. This bit is set to a one if the ATU cannot load or learn a new mapping due to all the available locations for an address being static or if an ATU Violation is detected. It is automatically cleared when all the pending ATU Violations have been serviced by the ATU Get/Clear Violation Data operation (Table 110). This bit being high causes the device's INTn pin to go low if the ATUProbIntEn bit in Global Control (Table 101) is set to a one.

Table 96: Switch Global Status Register (Continued)
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
2	ATUDone	LH	ATU Done Interrupt. This bit is set to a one whenever the ATUBusy bit (Table 110) transitions from a one to a zero. It is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the ATUDoneIntEn bit in Global Control (Table 100) is set to a one.
1	PHYInt	RO	PHY Interrupt. This bit is set to a one when the internal PHYs interrupt logic has at least one active interrupt (from ports 0 to 7). This bit being high causes the device's INTn pin to go low if the PHYIntEn bit in Global Control (Table 100) is set to a one. This bit does not clear until all the PHY interrupts are serviced.
0	EEInt	LH	EEPROM Done Interrupt. This bit is set to a one after the EEPROM is done loading registers and it is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the EEIntEn bit in Global Control (Table 100) is set to a one.

Table 97: ATU FID Register
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:0	FID	RWR	ATU MAC Address Forwarding Information Database number. If multiple address databases are not being used these bits must remain zero. If multiple address databases are being used these bits are used to set the desired address database number that is to be used on the Database supported commands (ATUOps 0x3, 0x4, 0x5 and 0x6 above). On Get/Clear Violation Data ATUOps these bits return the FID] value associated with the ATU violation that was just serviced.

Table 98: VTU FID Register
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15:13	Reserved	RES	Reserved for future use.
12	VIDPolicy	RWR	VID Policy. This bit is used to indicate any frames associated with this VID value are to be trapped to the TrapDest port (global offset 0x1A), monitored to the MirrorDest port (global offset 0x1A) or discarded. The action that takes place is determined by the frame's ingressing port's VTUPolicy bits (port offset 0x0E).
11:0	FID	RWR	VTU MAC Address Forwarding Information Database (FID) number. On VTU Load and VTU GetNext operations, this field is VTU FID and it is used to separate MAC address databases by a frame's VID. If multiple address databases are not being used these bits must remain zero. If multiple address databases are being used these bits are used to set the desired address database number that is associated with a VID value on Load operations (or these bits are used to return the currently assigned FID value found in the VTU on Get Next operations).
11:6	Reserved	RES	Reserved for future use.

Table 99: VTU SID Register
Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:6	Reserved	RES	Reserved for future use.
5:0	SID	RWR	VTU 802.1s Port State Information Database (SID) number. On VTU Load and VTU GetNext operations this field is the SID data that is associated with the VID that is being loaded or read in the VTU. If 802.1s multiple spanning trees are not being used these SID bits must remain zero. If multiple spanning trees are being used these bits are used to define the desired 802.1s information database (SID) number that is associated with the VID value on Load operations (or these bits are used to return the currently assigned SID value found in the VTU on Get Next operations). On STU Load and STU GetNext operations this field is used as the SID that is associated with the STU data (Global 1, offsets 0x07 to 0x09).

Table 100: Switch Global Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15	SWReset	SC	Switch Software Reset. Writing a one to this bit causes the QC, the MAC state machines in the switch to be reset. Register values are not modified. The EEPROM is not re-read. The ATU, VTU, MIBs, PHYs are not affected by this bit. When the reset operation is complete, this bit is cleared to a zero automatically. The reset occurs immediately. To prevent transmission of CRC frames, set all of the ports to the Disabled state (Table 76), and wait for 2 ms. (i.e., the time for a maximum frame to be transmitted at 10 Mbps) before setting the SWReset bit to a one.
14	Reserved	RES	Reserved for future use.
13	Discard Excessive	RWR	Discard frames with Excessive Collisions. When this bit is set to a one frames that encounter 16 consecutive collisions are discarded. When this bit is cleared to a zero Egress frames are never discarded and the backoff range is reset after 16 consecutive collisions on a singe frame.
12	ARPwoBC	RWR	ARP detection without Broadcast checking. When enabled the switch core does not check for a Broadcast MAC address as part of the ARP frame detection. It only checks the Ether Type and makes the decision. When disabled the switch core checks for both the Ether Type and Broadcast MAC address for ARP frame detection.
11	Reserved	RES	Reserved for future use.
10	MaxFrame Size	RWR	Maximum Frame Size allowed. The Ingress block discards all frames less than 64 bytes in size. It also discards all frames greater than a certain size (regardless of whether or not the frame is IEEE 802.3ac tagged) as follows: 0 = Max size is 1522 for IEEE tagged frames, 1518 for untagged frames 1 = Max size is 1632
9	ReLoad	SC	Reload the registers using the EEPROM. When this bit is set to a one, the contents of the external EEPROM are used to load the registers just as if a reset had occurred. When the reload operation finishes, this bit is cleared to a zero automatically and the EEInt interrupt bit is set.
8	Reserved	RES	Reserved for future use.
7	DevIntEn	RWR	Device Interrupt Enable. This bit must be set to a one to allow the Device interrupt to drive the device's INTn pin low.
6	StatsDone IntEn	RWR	Statistics Operation Done Interrupt Enable. This bit must be set to a one to allow the Stat Done interrupt to drive the device's INTn pin low.
5	VTUProb IntEn	RWR	VLAN Problem/Violation Interrupt Enable. This bit must be set to a one to allow the VTUProblem interrupt to drive the device's INTn pin low.
4	VTUDone IntEn	RWR	VLAN Table Operation Done Interrupt Enable. This bit must be set to a one to allow the VTUDone interrupt to drive the device's INTn pin low.

Table 100: Switch Global Control Register (Continued)
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
3	ATUProb IntEn	RWR	ATU Problem/Violation Interrupt Enable. This bit must be set to a one to allow the ATU Problem interrupt to drive the device's INTn pin low.
2	ATUDone IntEn	RWR	ATU Operation Done Interrupt Enable. This bit must be set to a one to allow the ATU Done interrupt to drive the device's INTn pin low.
1	PHYIntEn	RWR	PHY Interrupt Enable. This bit must be set to a one to allow active interrupts enabled in PHY register 0x12 to drive the device's INTn pin low.
0	EEIntEn	RWS	EEPROM Done Interrupt Enable. This bit must be set to a one to allow the EEPROM Done interrupt to drive the device's INTn pin low.

Table 101: VTU Operation Register
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	VTUBusy	SC	VLAN Table Unit Busy. This bit must be set to a one to start a VTU operation (see VTUOp below). Only one VTU operation can be executing at one time so this bit must be zero before setting it to a one. When the requested VTU operation completes, this bit will automatically be cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (Table 100).
14:12	VTUOp	RWR	VLAN Table Unit Table Opcode. The devices support the following VTU operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Flush All Entries in the VTU and STU 010 = No Operation 011 = VTU Load[1] or Purge[2] an Entry 100 = VTU Get Next[3] 101 = STU Load[4] or Purge[5] an Entry 110 = STU Get Next[6] 111 = Get/Clear Violation Data[7]
11:7	Reserved	RES	Reserved for future use
6	Member Violation	RO	Source Member Violation. On Get/Clear Violation Data VTUOps, this bit is returned set to a one if the Violation being serviced is due to an 802.1Q Member Violation. A Member Violation occurs when an 802.1Q enabled Ingress port accesses the VTU with a VID that is contained in the VTU but whose Membership list does not include this Ingress port. Only the first Member Violation or Miss Violation (below) will be saved until cleared.
5	Miss Violation	RO	VTU Miss Violation. On Get/Clear Violation Data VTUOps this bit is returned set to a one if the Violation being serviced was due to an 802.1Q Miss Violation. A Miss Violation occurs when an 802.1Q enabled Ingress port accesses the VTU with a VID that is not contained in the VTU. Only the first Miss Violation or Member Violation (above) is saved until cleared.
4	Reserved	RES	Reserved for future use

Table 101: VTU Operation Register (Continued)
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
3:0	SPID	RO	On the Get Violation Data VTUOp, this field returns the Source Port ID of the port that caused the violation. If SPID 0xF the source of the violations was the CPU register interface (i.e., the VTU was full during a CPU Load operation).

1. A VTU Entry is Loaded when the Valid bit (in the VTU VID register at global offset 0x06) is set to a one. This VTU Load is the only VTUOp that uses the FID & SID field and it uses them as data to be loaded along with the desired VID and its port member data.
2. An VTU Entry is Purged when the Valid bit (in the VTU FID register at global offset 0x06) is cleared to a zero.
3. A VTU Get Next operation finds the next higher VID currently in the VTU's database. The VID value (Table 102) is used as the VID to start from. To find the lowest VID set the VID field to ones. When the operation is done the VID field contains the next higher VID currently active in the VTU. To find the next VID simply issue the VTU Get Next opcode again. If the VID field is returned set to all one's with the Valid bit cleared to zero, no higher VID's were found. To Search for a particular VID, perform a VTU Get Next operation using a VID field with a value one less than the one being searched for.
4. A SID Entry is Loaded if the Valid bit (in the VTU VID register at global offset 0x06) is set to a one. This STU Load uses the SID as a pointer into the SID Translation Unit (STU). The data loaded into the STU is the lower two bits of each port's VTU Data that are used to define the 802.1s port states that are to be associated with this SID.
5. A SID Entry is Purged if it exists and the Valid bit (in the VTU VID register at global offset 0x06) is cleared to a zero.
6. A STU Get Next operation finds the next higher SID currently in the STU's database. The SID value is used as the SID to start from. To find the lowest SID set the SID field to ones. When the operation is done the SID field contains the next higher SID currently active in the STU. To find the next SID simply issue the STU Get Next opcode again. If the SID field is returned set to all one's with a Valid bit cleared to zero, no higher SID's were found. To Search for a particular SID, perform a STU Get Next operation using a SID field with a value one less than the one being searched for.
7. When the VTUProb bits is set to a one (Global Status—Table 96) the Get/Clear Violation VTUOp can be used to retrieve the data associated with the Violation. It will return the source port of the violation in the SPID field of this registers (bits 3:0) and it will return the VID of the violation in the VID field of the VTU VID register (Table 102). When all Violations currently pending in the VTU have been serviced the VTUProb bit in Global Status will be cleared to a zero.

Table 102: VTU VID Register
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description
15:13	Reserved	RES	Reserved for future use.
12	Valid	RWR	Entry's Valid bit. At the end of VTU (or STU) Get Next operations, if this bit is set to a one it indicates the VID (or SID) value below is valid (or the SID value above is valid). If this bit is cleared to a zero and the VID (or SID) is all ones, it indicates the end of the VID (or SID) list was reached with no new valid entries found. On Load or Purge operations, this bit indicates the desired operation of a Load (when set to a one) or a Purge (when cleared to a zero).
11:0	VID	RWR	VLAN Identifier. This VID is used in the VTU Load or VTU GetNext operation and it is the VID that is associated with the VTU data below (Table 103) or the VID that caused the VTU Violation.

Table 103: VTU/STU Data Register Ports 0 to 3 for VTU Operations
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
13:12	Member TagP3	RWR	Membership and Egress Tagging for Port 3. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP0 below.
11:10	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
9:8	Member TagP2	RWR	Membership and Egress Tagging for Port 2. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP0 below.
7:6	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
5:4	Member TagP1	RWR	Membership and Egress Tagging for Port 1. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP0 below.
3:2	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
1:0	Member TagP0	RWR	Membership and Egress Tagging for Port 0. These bits are used to support 802.1Q membership and Egress Tagging as follows: 00 = Port is a member of this VLAN and frames are to egress unmodified. 01 = Port is a member of this VLAN and frames are to egress Untagged. 10 = Port is a member of this VLAN and frames are to egress Tagged. 11 = Port is not a member of this VLAN. Any frames with this VID[1] are discarded at ingress and are not allowed to egress this port.

1. The VID used comes from the VID in Tagged frames or the default VID assigned to Untagged frames.

Table 104: VTU/STU Data Register Ports 0 to 3 for STU Operations
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:14	PortState P3	RWR	Per VLAN Port States for Port 3. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP0 below.
13:12	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
11:10	PortState P2	RWR	Per VLAN Port States for Port 2. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP0 below.
9:8	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
7:6	PortState P1	RES	Per VLAN Port States for Port 1. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP0 below.
5:4	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
3:2	PortState P0	RES	Per VLAN Port States for Port 0. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. The Per VLAN Port States are: 00 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 01 = Blocking/Listening Port State for this port for frames with this VID. 10 = Learning Port State for this port for frames with this VID. 11 = Forwarding Port State for this port for frames with this VID.
1:0	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.

Table 105: VTU/STU Data Register Ports 4 to 7 for VTU Operations
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
13:12	Member TagP7	RWR	Membership and Egress Tagging for Port 7. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP4 below.
11:10	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
9:8	Member TagP6	RWR	Membership and Egress Tagging for Port 6. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP4 below.
7:6	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
5:4	Member TagP5	RWR	Membership and Egress Tagging for Port 5. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP4 below.
3:2	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
1:0	Member TagP4	RWR	Membership and Egress Tagging for Port 4. These bits are used to support 802.1Q membership and Egress Tagging as follows: 00 = Port is a member of this VLAN and frames are to egress unmodified. 01 = Port is a member of this VLAN and frames are to egress Untagged. 10 = Port is a member of this VLAN and frames are to egress Tagged. 11 = Port is not a member of this VLAN. Any frames with this VID[1] are discarded at ingress and are not allowed to egress this port.

1. The VID used comes from the VID in Tagged frames or the default VID assigned to Untagged frames.

Table 106: VTU/STU Data Register Ports 4 to 7 for STU Operations
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15:14	PortState P7	RWR	Per VLAN Port States for Port 7. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP4 below.
13:12	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
11:10	PortState P6	RWR	Per VLAN Port States for Port 6. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP4 below.
9:8	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
7:6	PortState P5	RES	Per VLAN Port States for Port 5. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP4 below.
5:4	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
3:2	PortState P4	RES	Per VLAN Port States for Port 4. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. The Per VLAN Port States are: 00 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 01 = Blocking/Listening Port State for this port for frames with this VID. 10 = Learning Port State for this port for frames with this VID. 11 = Forwarding Port State for this port for frames with this VID.
5:4	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.

Table 107: VTU/STU Data Register Port 8 to 10 for VTU Operations
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15	VIDPRI Override	RWR	VID Priority Override. When this bit is set to a one the VIDPRI bits (below) are used to override the priority on any frame associated with this VID.
14:12	VIDPRI	RWR	VID Priority bits. These bits are used to override the priority on any frames associated with this VID value, if the VIDPRIOverride bit (above) is set to a one.
11:10	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
9:8	MemberTagP10	RWR	Membership and Egress Tagging for Port 10. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP8 below.
7:6	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
5:4	MemberTagP9	RWR	Membership and Egress Tagging for Port 9. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP8 below.

Table 107: VTU/STU Data Register Port 8 to 10 for VTU Operations
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
3:2	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
1:0	MemberTagP8	RWR	Membership and Egress Tagging for Port 8. These bits are used to support 802.1Q membership and Egress Tagging as follows: 00 = Port is a member of this VLAN and frames are to egress unmodified. 01 = Port is a member of this VLAN and frames are to egress Untagged. 10 = Port is a member of this VLAN and frames are to egress Tagged. 11 = Port is not a member of this VLAN. Any frames with this VID[1] are discarded at ingress and are not allowed to egress this port.

1. The VID used is the VID assigned to the frame during ingress.

Table 108: VTU/STU Data Register Port 8 to 10 for STU Operations
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:10	PortState P10	RWR	Per VLAN Port States for Port 10. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP8 below.
9:8	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
7:6	PortState P9	RWR	Per VLAN Port States for Port 9. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP8 below.
5:4	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
3:2	PortState P8	RWR	Per VLAN Port States for Port 8. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. The Per VLAN Port States are: 00 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 01 = Blocking/Listening Port State for this port for frames with this VID. 10 = Learning Port State for this port for frames with this VID. 11 = Forwarding Port State for this port for frames with this VID.
1:0	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.

Table 109: ATU Control Register
Offset: 0x0A or Decimal 10

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:4	AgeTime	RWS to 0x16	ATU Age Time. These bits determine the time that each ATU Entry remains valid in the database, since its last access as a source address, before being purged. The value in this register times 15 is the age time in seconds. For example: The default value of 0x16 is 22 decimal. 22 x 15 = 330 seconds or 5.5 minutes, which results in an average age time of 306 seconds, or about 5 minutes (as this register setting sets the maximum time with the minimum time being the Max. Time - Max. Time/7, and the average time being the average between those two times). The minimum age time is 0x1 or 15 seconds. The maximum age time is 0xFF or 3825 seconds or almost 64 minutes. If the AgeTime is set to 0x0 the Aging function is disabled and all learned addresses will remain in the database forever.
3	Learn2All	RWR or RWS[1]	Learn to All devices in a Switch. When more than one Marvell® device is used to for a single 'switch' it may be desirable for all devices in the 'switch' to learn any address this device learns[2]. When this bit is set to a one all other devices in the 'switch' learn the same addresses this device learns. When this bit is cleared to a zero only the devices that actually receive frames will learn from those frames. This mode typically supports more active MAC addresses at one time as each device in the switch does not need to learn addresses it may never use. Learn2All must be set to a 1 when hardware learn limiting is enabled on any port in the device (port offset 0x0C).
2:0	Reserved	RES	Reserved for future use.

1. Learn2All is initialized set if both the P9_TXD3/SW_24P configuration pin and the MDC_PHY/PPU_EN/LRN2ALL configuration pin are high at the rising edge of RESETn.
2. Learn2All message learning frames will be sent out a port if that port's MessagePort bit is set to a one (Table 77). If this frame is used it is recommended that all DSA Tag ports, except for the CPU's port, have their MessagePort but set to a one. Ports that are not DSA Tag ports (i.e., normal Network ports) should not have their MessagePort bit set to a one.

Table 110: ATU Operation Register
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
15	ATUBusy	SC	Address Translation Unit Busy. This bit must be set to a one to start an ATU operation (see ATUOp below). Only one ATU operation can be executing at one time so this bit must be zero before setting it to a one. When the requested ATU operation completes, this bit is automatically be cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (Table 100).
14:12	ATUOp	RWR	Address Translation Unit Opcode. The devices support the following ATU operations. (All of these operations can be executed while frames are passing through the switch): 000 = No Operation 001 = Flush[1]/Move[2] All Entries 010 = Flush[3]/Move all Non-Static[4] Entries 011 = Load[5] or Purge[6] an Entry in a particular FID Database 100 = Get Next[7] from a particular FID Database 101 = Flush/Move All Entries in a particular FID Database 110 = Flush/Move all Non-Static Entries in a particular FID Database 111 = Get/Clear Violation Data[8]
11	Reserved	RES	Reserved for future use.
10:8	MACPri	RWR	MAC Priority bits. These bits are used to override the priority on any frames associated with this MAC value, if the EntryState bits indicate MAC Priority can be used – see Section 8.8.1) and the port's SA and/or DA priority overrides are enabled (in Port Control 2 – Table 80).
7	AgeOut Violation	RO	Age Out Violation. On Get/Clear Violation Data ATUOps this bit is returned set to a one if the Violation being serviced was due to a non-static entry being aged with an EntryState = 0x1. AgeOutViolations will only occur on entries that are associated with ports whose IntOnAgeOut bit is set to a one (port offset 0x0B). Up to 2 Age Out Violations will be saved per device until cleared. An Age Out Violation will return the violating MAC in global registers at offset 0x0D, 0x0E and 0x0F. The ATU Data Register at global offset 0x0C will contain the violating MAC's Trunk bit, its DPV or Trunk ID and its Entry State. The violating MAC's PRI bits will be updated in MACPri (global offset 0x0B) and it BIN will be updated in global offset 0x06).
6	Member Violation	RO	Source Port Violation. On Get/Clear Violation Data ATUOps this bit is returned set to a one if the Violation being serviced is due to a Source Address look-up that resulted in a Hit but where the ATUData[8:0] bits does not contain the frame's Ingress port bit set to a one (i.e., a station move occurred). This violation can be masked on a per port basis by setting the port's IngoreWrongData bit. Only the first Member Violation, Miss Violation (below) or Full Violation (below) is be saved per port until cleared.

Table 110: ATU Operation Register (Continued)
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
5	Miss Violation	RO	ATU Miss Violation. On Get/Clear Violation Data ATUOps this bit is returned set to a one if the Violation being serviced is due to a Source Address look-up that resulted in a Miss on ports that are Locked (i.e., CPU directed learning is enabled on the port). If Age Violations are enabled (ATUAgeIntEn = 1 in global 2, offset 0x05) and Locked ports are not allowed to self refresh addresses (RefreshLocked = 0 in port offset 0x0B) this Miss Violation will also occur if the frame's Source Address was found in the address database with an EntryState less than 0x4 (i.e., it is about half way aged out). Only the first Miss Violation, Member Violation (above) or Full Violation (below) is saved per port until cleared.
4	ATUFull Violation	RO	ATU Full Violation. On Get/Clear Violation Data ATUOps this bit is set to a one if the Violation being serviced is due to a Load ATUOp or automatic learn that could not store the desired entry. This only occurs if all available locations for the desired address contain other MAC addresses that are loaded Static. Only the first Full Violation, Member Violation (above) or Miss Violation (above) is saved per port until cleared.
3:0	Reserved	RES	Reserved for future use.

1. A Flush occurs when the EntryState (Table 111) is zero.
2. Move is used for 802.1X (rapid spanning tree) to reassign all valid entries associated with one port (the FromPort - Table 111) and move the association to another port (the ToPort - Table 111). It can also be used to completely disassociate a port from the database (if the ToPort = 0xF). The Move occurs if the EntryState (Table 111) is 0xF.
3. A Non-Static entry is any unicast address with an EntryState less than 0x8. All unicast frames will flood until new addresses are learned.
4. A Non-Static entry is any unicast address with an EntryState less than 0x8. All unicast frames flood until new addresses are learned.
5. An Entry is Loaded when the EntryState (Table 111) is non-zero.
6. An Entry is Purged when the EntryState (Table 111) is zero.
7. A Get Next operation finds the next higher MAC address currently in a particular ATU database (defined by the FID field - Global offset 0x01). The ATUByte[5:0] values (Table 112) are used as the starting address. To find the lowest MAC address set ATU[5:0] to ones. When the operation is done, ATUByte[5:0] contains the next higher MAC address. To find the next address, simply issue the Get Next opcode again. If ATUByte[5:0] is returned set to all one's with an EntryState of 0x0, no higher MAC address was found. If ATUByte[5:0] is returned set to all one's with a non-zero EntryState, the highest MAC address was found (i.e., the Broadcast address) and the end of the table was reached. To search for a particular address, perform a Get Next operation using a MAC address with a value one less than the one being searched for.
8. When the ATUProb bit is set to a one (Global Status - Table 96), the Get/Clear Violation ATUOp can be used to retrieved the data associated with the violation. When all violations currently pending in the ATU have been serviced the ATUProb bit in the Global Status is cleared to a zero.

Table 111: ATU Data Register
Offset: 0x0C or Decimal 12

Bits	Field	Type	Description
15	Trunk	RWR	Trunk Mapped Address. When this bit is set to a one the data bits 7:4 below (PortVec bits [3:0]) is the Trunk ID assigned to this address. PortVec bits [10:4] must be written as zero when this bit is set to a one. When this bit is cleared to a zero the data in bits 14:4 below (PortVec bit[10:0]) is the port vector assigned to this address.
14:4	PortVec/ ToPort & FromPort	RWR	Port Vector. If the Trunk bit, above, is zero, these bits are used as the input Port Vector for ATU Load operations and it's the resulting Port Vector from ATU Get Next operations. The lower four bits (7:4) are used as the FromPort and the next higher four bits (11:8) are used as the ToPort during move operations. If the ToPort = 0xF, the operation becomes a RemovePort (i.e., the FromPort is removed from the database and the entry is purged if the resulting PortVec equals zeros).
3:0	EntryState/ SPID	RWR	ATU Entry State. These bits are used as the Entry State for ATU Load/ Purge or Flush/Move operations and it is the resulting Entry State from ATU Get Next operations (GetNext is the only ATU operation supported in the devices). If these bits equal 0x0 then the ATUOp is a Purge or a Flush. If these bits are not 0x0 then the ATUOp is a Load or a Move (a Move ATUOp requires these bits to be 0xF). On Get/Clear Violation Data ATUOps, these bits return the Source Port ID (SPID) associated with the ATU violation that was just serviced, except for Age Out violation where these return 0x1. If SPID = 0xF the source of the violation was the CPU's register interface (i.e., the ATU was full during a CPU Load operation).

- The ATU Entry State bits on Unicast ATU entries are defined as follows:
- 0x0: Unused entry
- 0x1 to 0x7: Used entry where Entry State = the Age of the entry where 0x1 is the oldest
- 0x8: Static Policy entry
- 0x9: Static Policy entry with Priority Override
- 0xA: Static Non Rate Limiting (NRL) entry
- 0xB: Static Non Rate Limiting (NRL) entry with Prority Override
- 0xC: Static entry defining frames with this DA as MGMT
- 0xD: Static entry defining frames with this DA as MGMT with Priority Override
- 0xE: Static entry
- 0xF: Static entry with Priority Override

The ATU Entry State bits on Multicast ATU entries are defined as follows:

- 0x0: Unused entry
- 0x1 to 0x3: Reserved for future use
- 0x4: Static Policy entry
- 0x5: Static Non Rate Limiting (NRL) entry
- 0x6: Static entry defining frames with this DA as MGMT
- 0x7: Static entry
- 0x8 to 0xB: Reserved for future use
- 0xC: Static Policy entry with Priority Override
- 0xD: Static Non Rate Limiting (NRL) entry with Priority Override
- 0xE: Static entry defining frames with this DA as MGMT with Priority Override
- 0xF: Static entry with Priority Override

Table 112: ATU MAC Address Register Bytes 0 & 1
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
15:8	ATUByte0	RWR	ATU MAC Address Byte 0 (bits 47:40) used as the MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. Bit 0 of byte 0 (bit 40) is the multicast bit (it is the first bit down the wire). Any MAC address with the multicast bit set to a one is considered Static by the ATU. On Get/Clear Violation Data ATUOps these bits return ATUByte0 associated with the ATU violation that was just serviced.
7:0	ATUByte1	RWR	ATU MAC Address Byte 1 (bits 39:32) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte1 associated with the ATU violation that was just serviced.

Table 113: ATU MAC Address Register Bytes 2 & 3
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
15:8	ATUByte2	RWR	ATU MAC Address Byte 2 (bits 31:24) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte2 associated with the ATU violation that was just serviced.
7:0	ATUByte3	RWR	ATU MAC Address Byte 3 (bits 23:16) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte3 associated with the ATU violation that was just serviced.

Table 114: ATU MAC Address Register Bytes 4 & 5
Offset: 0x0F or decimal 15

Bits	Field	Type	Description
15:8	ATUByte4	RWR	ATU MAC Address Byte 4 (bits 15:8) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte4 associated with the ATU violation that was just serviced.
7:0	ATUByte5	RWR	ATU MAC Address Byte 5 (bits 7:0) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte5 associated with the ATU violation that was just serviced.

Table 115: IP-PRI Mapping Register 0
Offset: 0x10 or Decimal 16

Bits	Field	Type	Description
15:14	IP_0x1C	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x1C.
13:12	IP_0x18	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x18.
11:10	IP_0x14	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x14.
9:8	IP_0x10	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x10.
7:6	IP_0x0C	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x0C.
5:4	IP_0x08	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x08.
3:2	IP_0x04	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x04.
1:0	IP_0x00	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x00.

Table 116: IP-PRI Mapping Register 1
Offset: 0x11 or Decimal 17

Bits	Field	Type	Description
15:14	IP_0x3C	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x3C.
13:12	IP_0x38	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x38.
11:10	IP_0x34	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x34.
9:8	IP_0x30	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x30.
7:6	IP_0x2C	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x2C.
5:4	IP_0x28	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x28.
3:2	IP_0x24	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x24.
1:0	IP_0x20	RWR	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x20.

Table 117: IP-PRI Mapping Register 2
Offset: 0x12 or Decimal 18

Bits	Field	Type	Description
15:14	IP_0x5C	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x5C.
13:12	IP_0x58	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x58.
11:10	IP_0x54	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x54.
9:8	IP_0x50	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x50.
7:6	IP_0x4C	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x4C.
5:4	IP_0x48	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x48.
3:2	IP_0x44	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x44.
1:0	IP_0x40	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x40.

Table 118: IP-PRI Mapping Register 3
Offset: 0x13 or Decimal 19

Bits	Field	Type	Description
15:14	IP_0x7C	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x7C.
13:12	IP_0x78	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x78.
11:10	IP_0x74	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x74.
9:8	IP_0x70	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x70.
7:6	IP_0x6C	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x6C.
5:4	IP_0x68	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x68.
3:2	IP_0x64	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x64.
1:0	IP_0x60	RWS to 0x1	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x60.

Table 119: IP-PRI Mapping Register 4
Offset: 0x14 or Decimal 20

Bits	Field	Type	Description
15:14	IP_0x9C	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x9C.
13:12	IP_0x98	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x98.
11:10	IP_0x94	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x94.
9:8	IP_0x90	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x90.
7:6	IP_0x8C	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x8C.
5:4	IP_0x88	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x88.
3:2	IP_0x84	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x84.
1:0	IP_0x80	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0x80.

Table 120: IP-PRI Mapping Register 5
Offset: 0x15 or Decimal 21

Bits	Field	Type	Description
15:14	IP_0xBC	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xBC.
13:12	IP_0xB8	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xB8.
11:10	IP_0xB4	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xB4.
9:8	IP_0xB0	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xB0.
7:6	IP_0xAC	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xAC.
5:4	IP_0xA8	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xA8.
3:2	IP_0xA4	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xA4.
1:0	IP_0xA0	RWS to 0x2	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xA0.

Table 121: IP-PRI Mapping Register 6
Offset: 0x16 or Decimal 22

Bits	Field	Type	Description
15:14	IP_0xDC	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xDC.
13:12	IP_0xD8	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xD8.
11:10	IP_0xD4	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xD4.
9:8	IP_0xD0	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xD0.
7:6	IP_0xCC	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xCC.
5:4	IP_0xC8	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xC8.
3:2	IP_0xC4	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xC4.
1:0	IP_0xC0	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xC0.

Table 122: IP-PRI Mapping Register 7
Offset: 0x17 or Decimal 23

Bits	Field	Type	Description
15:14	IP_0xFC	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xFC.
13:12	IP_0xF8	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xF8.
11:10	IP_0xF4	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xF4.
9:8	IP_0xF0	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xF0.
7:6	IP_0xEC	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xEC.
5:4	IP_0xE8	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xE8.
3:2	IP_0xE4	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xE4.
1:0	IP_0xE0	RWS to 0x3	IPv4 and IPv6 mapping. The value in this field is used as the frame's priority if bits (7:2) of its IP TOS/DiffServ/Traffic Class value is 0xE0.

Table 123: IEEE-PRI Register
Offset: 0x18 or Decimal 24

Bits	Field	Type	Description
15:14	Tag_0x7	RWS to 0x3	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 7.
13:12	Tag_0x6	RWS to 0x3	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 6.
11:10	Tag_0x5	RWS to 0x2	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 5.
9:8	Tag_0x4	RWS to 0x2	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 4.
7:6	Tag_0x3	RWS to 0x1	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 3.
5:4	Tag_0x2	RWS to 0x1	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 2.
3:2	Tag_0x1	RWR	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 1.
1:0	Tag_0x0	RWR	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 0.

Table 124: Monitor Control
Offset: 0x1A or decimal 26

Bits	Field	Type	Description
15:12	Ingress Monitor Dest	RWS	Ingress Monitor Destination Port. Frames that are targeted toward an Ingress Monitor Destination go out the port number indicated in these bits. This includes frames received on a DSA Tag port with the Ingress Monitor type, and frames received on a Network port that is enabled to be the Ingress Monitor Source Port (Table 76). If the Ingress Monitor Destination Port resides in this device these bits should point to the Network port where these frames are to egress. If the Ingress Monitor Destination Port resides in another device these bits should point to the DSA Tag port in this device that is used to get to the device that contains the Ingress Monitor Destination Port.
11:8	Egress Monitor Dest	RWS	Egress Monitor Destination Port. Frames that are targeted toward an Egress Monitor Destination go out of the port number indicated in these bits. This includes frames received on a DSAI Tag port with the Egress Monitor type, and frames transmitted on a Network port that is enabled to be the Egress Monitor Source Port (Table 80). If the Egress Monitor Destination port resides in this device, these bits should point to the Network port where these frames are to egress. If the Egress Monitor Destination Port resides in another device, these bits should point to the DSA Tag port in this device that is used to reach the device that contains the Egress Monitor Destination Port.

Table 124: Monitor Control
Offset: 0x1A or decimal 26

Bits	Field	Type	Description
7:4	CPU Dest	RWS	CPU Destination Port. Many modes of frame processing need to know where the CPU is located. These modes are: 1. When IGMP/MLD frame is received and Snooping is enabled on the port (port offset 0x04) 2. When this port is configured as a DSA Port and it receives a To_Cpu frame[1] 3. When a Rsvd2CPU frames enters the port (global 2 offset 0x05) 4. When the port's SA Filtering mode is Drop to CPU (port offset 0x04) 5. When any of the port's Policy Options (port offset 0x0E) trap the frame to the CPU 6. When the ingressing frame is a ARP and ARP mirroring is enabled in the device (port offset 0x08) In all cases, except for ARP, the frames that meet the enabled criteria are mapped to the port defined by this register only, overriding where the frame would normally go. In the case of ARP the frame will be mapped normally and it will also get copied to this port. Frames that filtered or discarded will not be mapped to the CPUDest with the exception of the Rsvd2CPU and DSA Tag cases (numbers 2 and 3). The CPUDest bits indicate the port number on this device where the CPU is connected (either directly or indirectly through another Marvell® switch device). If CPUDest = 0xF the remapped frames will be discarded, no ARP mirroring will occur and ingressing To_CPU frames will be discarded. Note: MGMT or BPDU frames detected by using the ATU are directed to the correct port where the CPU is connected by ensuring the CPU port's bit is set in the frame's MGMT DA MAC address as stored in the ATU address database (Section 4.4.5).
3:0	Mirror Dest	RWS	Mirror Destination Port. Frames that ingress a port that trigger a policy mirror are mapped (copied) to this port as long as the frame is not filtered or discarded. The MirrorDest should point to the port that directs these frames to the CPU that will process these frames. This target port should be a DSA Tag port so the frames will egress with a To_CPU DSA Tag with a CPU Code of Policy Mirror. To_CPU DSA Tag frames with a CPU Code of Policy Mirror that ingress a DSA Tag port will be sent to the port number defined in MirrorDest. If MirrorDest = 0xF Policy Mirroring is disabled and ingressing To_CPU Policy Mirror frames will be discarded. The policy mirror enable bits are configurable per port (see Policy Control, port offset 0x0E).

1. To_CPU frames with a Code of Policy Mirror are mapped to the MirrorDest port (bits 3:0 of this register).

Table 125: Total Free Counter
Offset: 0x1B or Decimal 27

Bits	Field	Type	Description
15:9	Reserved	RES	Reserved for future use.
8:0	FreeQSize	RO	Free Queue Size Counter. This counter reflects the current number of unallocated buffers available for all the ports.

Table 126: Global Control 2
Offset: 0x1C or Decimal 28

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	DA Check	RWR	Check the DA on Remote Management frames. When this bit is set to a one the DA of Remote Management frames must be contained in this device's address database (ATU) as a Static entry (either unicast or multicast). If the DA of the frame is not contained in this device's address database the frame will not be processed as a Remote Management frame (i.e.,it will be discarded without further action if this device is the Trg_Dev of the frame). When this bit is cleared to zero the DA of Remote Management frames is not validated before processing the frame.
13	P10RM	RWS	Use Port 10 for Remote Management. When this bit is set to a one Remote Management frames will be accepted on and transmitted out Port 10 only. When this bit is cleared to a zero Remote Management will be accepted on and transmitted out Port 9 only (assuming Remote Management is enabled).
12	RMEnable	RWR	Remote Management Enable. When this bit is set to a one and this device receives a Remote Management Request frame directed to this device (on the port defined in bit 13 above) the frame will be processed and a Remote Management Response frame will be generated and sent out (the port defined in bit 13 above) if the DA of the frame matches the conditions of the DA Check bit above. In either case, the Request frame will be discarded (as it was directed to this device). When this bit is cleared to a zero Remote Management Request frames directed to this device will be discarded and ignored (i.e., it will not be processed and no Response frame will be generated). Regardless of the setting of this bit, Remote Management Request frames that are not directed to this device will be mapped to the port indicted by mapping the frame's Trg_Dev using the Device Mapping table (global 2, offset 0x06). Note: The setting of this bit will have no effect if the Remote Management port is in half-duplex mode. The port's FrameMode (port offset 0x05) must be DSA or EtherType DSA as well.
11:5	Reserved	RES	Reserved for future use.
4:0	DeviceNumber	RWS to 0xXX[1]	Device Number. In multi-chip systems, frames coming from a CPU (From_CPU frames) need to know when they have reached their destination chip. From_CPU frames whose Dev_Num field matches these bits have reached their destination chip and are sent out from this chip using the port number indicated in the frame's Trg_Port field. The DeviceNumber value must be unique for each chip in a Multi-chip system. These bits are set at reset by the ADDR[4:0] configuration pins.

1. The ADDR[4:0] configuration pins are used to set the initial value of this register. The ADDR[4:0] pins are also used to select between Multi-chip addressing mode or Single-chip addressing mode. Changing the value in this register after reset does *not* change the device's addressing mode nor its SMI address.

Table 127: Stats Operation Register
Offset: 0x1D or Decimal 29

Bits	Field	Type	Description
15	StatsBusy	SC	Statistics Unit Busy. This bit must be set to a one to start a Stats operation (See StatsOp below). Only one Stats operation can be executing at one time so this bit must be zero before setting it to a one. When the requested Stats operation completes, this bit automatically is cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (Table 100).
14:12	StatsOp	RWR	Statistics Unit Opcode. The devices support the following Stats operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Flush (clear) All Counters for all Ports 010 = Flush (clear) All Counters for a Port 011 = Reserved 100 = Read a Captured or Direct Counter 101 = Capture All Counters for a Port 11x = Reserved
11:10	Histogram Mode	RES to 0x3	Histogram Counters Mode. The Histogram mode bits control how the Histogram counters work as follows: 00 = Reserved 01 = Count received frames only 10 = Count transmitted frames only 11 = Count receive and transmitted frames
9	Reserved	RES	Reserved for future use

Table 127: Stats Operation Register (Continued)
Offset: 0x1D or Decimal 29

Bits	Field	Type	Description
8:5	StatsPort	RWR	Access Statistics Counters directly for a Port or the Capture area. These bits can be used to directly access a ports counters without doing a capture first. Use bits 8:5 = 0x0 to access the captured counters. Use bits 8:5 = 0x1 to access the counters for Port 0. Use bits 8:5 = 0x2 to access the counters for Port 1, etc. These bits must be zero for all StatsOps except for Read a Captured or Direct Counter command (e.g., these bits are not used for the Flush (clear) All Counters for a Port command).
5:0	StatsPtr	RWR	Statistics Pointer. This field is used as a parameter for the above StatsOp commands. It must be set to the desired Port number for the Capture All Counters for a Port (0x5) and Flush All Counters for a Port (0x2) StatsOps. Use 0x00 for Port 0, 0x01 for Port 1, etc. StatsPtr must be set to the desired counter to read for the Read a Captured Counter (0x4) StatsOp (valid range is 0x00 to 0x1F). A Capture All Counters for a Port StatsOp must be done prior to using the Read A Captured Counter StatsOp. The counter that is read is defined as follows: **Ingress Counters[1]** / **Egress Counters** 0x00 – InGoodOctetsLo / 0x0E – OutOctetsLo[2] 0x01 – InGoodOctetsHi / 0x0F – OutOctetsHi 0x02 – InBadOctets 0x04 – InUnicast / 0x10 – OutUnicast 0x06 – InBroadcasts / 0x13 – OutBroadcasts 0x07 – InMulticasts / 0x12 – OutMulticasts 0x16 – InPause / 0x15 – OutPause 0x18 – InUndersize 0x19 – InFragments 0x1A – InOversize 0x1B – InJabber 0x1C – In RxErr / 0x11 – Excessive 0x1D – InFCSErr / 0x1E – Collisions 0x05 – Deferred 0x14 – Single 0x17 – Multiple 0x03 – OutFCSErr 0x1F – Late **Histogram Counters[3]** 0x08 – 64Octets 0x09 – 65 to 127Octets 0x0A – 128 to 255Octets 0x0B – 256 to 511Octets 0x0C – 512 to 1023Octets 0x0D – 1024 to MaxOctets

1. If Marvell® Header mode is used on Ports 0 to 7 the extra two bytes in the frame are not included in the InGoodOctet nor the InBadOctet counts.
2. OutOctets may not accurately count the bytes transmitted on frames that encounter a collision.
3. If Marvell Header mode is used on Ports 0 to 7 the extra two bytes in the frame are not included in the count before determining which Histogram Counter to increment.

Table 128: Stats Counter Register Bytes 3 & 2
Offset: 0x1E or Decimal 30

Bits	Field	Type	Description
15:8	StatsByte3	RO	Statistics Counter Byte 3. These bits contain bits 31:24 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp— Table 127).
7:0	StatsByte2	RO	Statistics Counter Byte 2. These bits contain bits 23:16 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 127).

Table 129: Stats Counter Register Bytes 1 & 0
Offset: 0x1F or Decimal 31

Bits	Field	Type	Description
15:8	StatsByte1	RO	Statistics Counter Byte 1. These bits contain bits 15:8 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 127).
7:0	StatsByte0	RO	Statistics Counter Byte 0. These bits contain bits 7:0 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 127).

13.6 Switch Global 2 Registers

The devices contain a second set of global registers that effect all the Ethernet ports in the device. Each Global 2 register is 16-bits wide and their bit assignment are shown in Figure 75.

Figure 75: Global 2 Register bit Map (Device Addr 0x1C)

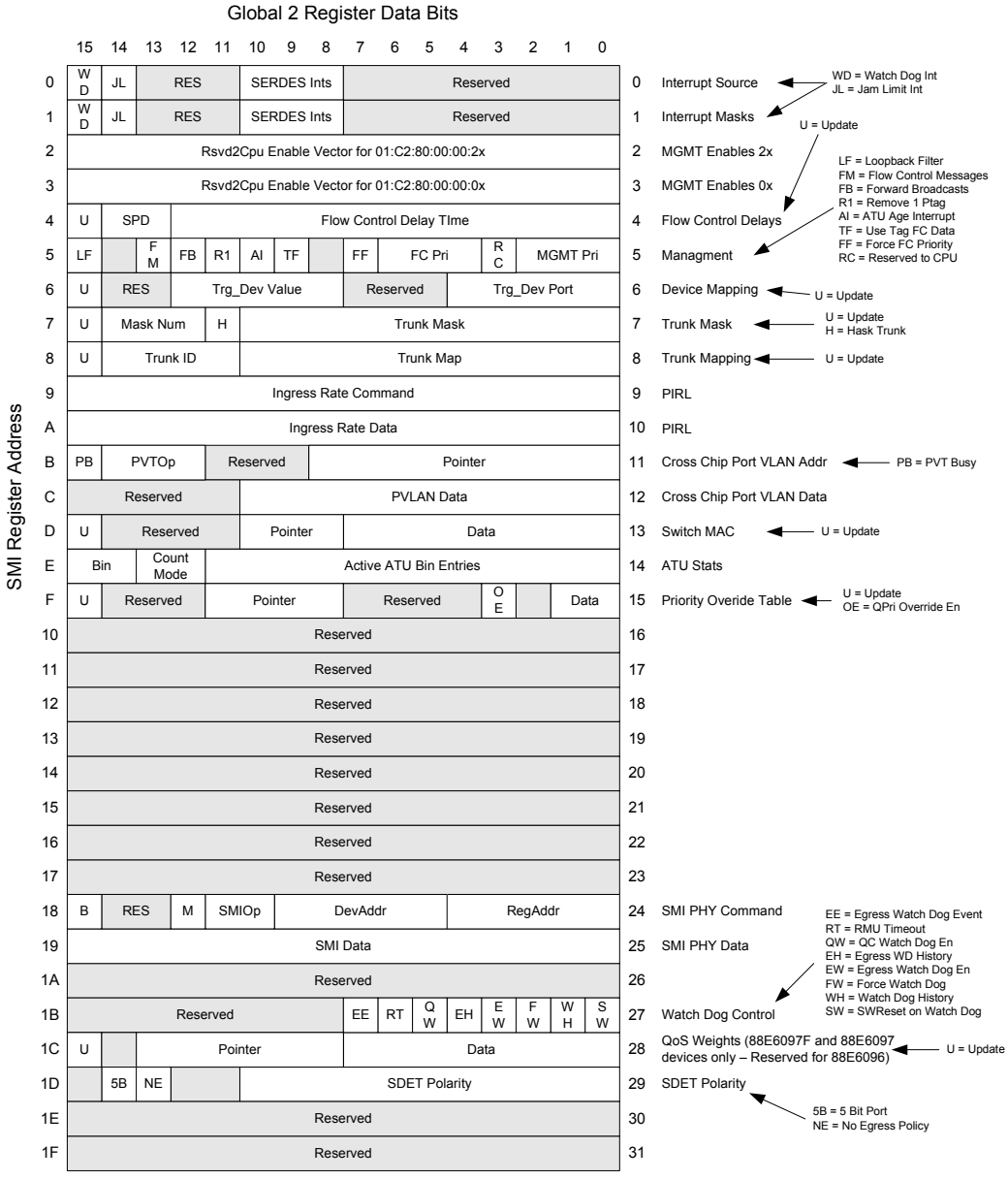


Table 130: Interrupt Source Register
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	WatchDog Int	ROC	WatchDog interrupt. This bit indicates a watch dog event occurred. Watch Dog events are enabled in the Watch Dog Control register (Global 2, offset 0x1B).
14	JamLimit	ROC	Jam Limit interrupt. This bit is set to a one when any of the ports detect an Ingress Jam Limit violation as determined by the port's LimitIn setting (in Jamming Control, port offset 0x02).
13:11	Reserved	RES	Reserved for future use.
10:8	SERDES LinkInt	ROC	SERDES interrupt for Ports 10 to 8. When the link changes state (link up or link down) on the SERDES this bit will be set to a one. These bits self clear on a read. Bit 10 is for port 10, bit 9 is for port 9, etc. These bits are only valid of the port is in 1000BASE-X mode.
7:0	Reserved	RES	Reserved for future use.

Table 131: Interrupt Mask Register
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15	WatchDog IntEn	RWR	WatchDog interrupt enable. This bit must be set to a one to allow the WatchDog interrupt (global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (global offset 0x00) so that the INTn pin can be driven low.
14	JamLimitEn	ROC	Jam Limit interrupt enable. This bit must be set to a one to allow the JamLimit interrupt (global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (global offset 0x00) so that the INTn pin can be driven low.
13:11	Reserved	RES	Reserved for future use.
10:8	SERDES LinkIntEn	ROC	SERDES interrupt enable for Ports 10 to 8. This bit must be set to a one to allow the SERDESLinkInt interrupt (global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (global offset 0x00) so that the INTn pin can be driven low.
7:0	Reserved	RES	Reserved for future use.

Table 132: MGMT Enable Register 2x
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15:0	Rsvd2CPU Enables 2x	RWS	Reserved DA Enables 2x. When the Rsvd2Cpu bit (Global 2, offset 0x05) is set to a one the 16 reserved multicast DA addresses whose bit in this register are also set to a one, are treated as MGMT[1] frames. The reserved DA's supported by this register take the form 01:80:C2:00:00:2x. When x = 0x0, bit 0 of this register is tested. When x = 0x2 bit 2 of this field is tested and so on with x = 0xF bit 15 of this register is tested. If the tested bit in this register is cleared to a zero, the frame will be treated as a normal (non-MGMT) frame. This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames. If the Rsvd2Cpu bit (Global 2, offset 0x05) is cleared to a zero these bits will have no effect.

1. MGMT, or management, frames are used for managed switch protocols like GVRP.

Table 133: MGMT Enable Register 0x
Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:0	Rsvd2CPU Enables 0x	RWS	Reserved DA Enables 0x. When the Rsvd2Cpu bit (Global 2, offset 0x05) is set to a one the 16 reserved multicast DA addresses whose bit in this register are also set to a one, are treated as MGMT[1] frames. All the reserved DA's supported by this register take the form 01:80:C2:00:00:0x. When x = 0x0, bit 0 of this register is tested. When x = 0x2 bit 2 of this field is tested and so on with x = 0xF bit 15 of this register is tested. If the tested bit in this register is cleared to a zero, the frame will be treated as a normal (non-MGMT) frame. This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames[2]. If the Rsvd2Cpu bit (Global 2, offset 0x05) is cleared to a zero these bits will have no effect.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregration (LAC). The switch processes MGMT frames differently (see Section 5.8).
2. Frames with a DA of 01:80:C2:00:00:01 (the Pause frame DA) are always treated as MAC control frames and cannot be treated as MGMT frames.

Table 134: Flow Control Delay Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15	Update	SC	Update FC Delay TIme data. When this bit is set to a one the data written to bits 12:0 will be loaded into the FC Delay Time register selected by the SPD bits below. After the write has taken place this bit self clears to zero.
14:13	SPD	RWR	Speed Number. These bits select one of three possible FC Delay Time register for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
12:0	FC Delay Time	RWS	Flow Control Delay Time. These bits are used to cause a MAC to assert Flow Control for the delay amount times 8.192 uSecs. The register used is determined by the Flow Control DSA Tag frame's SPD bits that was directed at this MAC. Three FC Delay Time registers are accessed by using the SPD bits above. SPD 0b00 is assigned for as the Flow Control delay to use when talking to 10 Mbit ports. SPD 0b01 is assigned for 100 Mbit ports and SPD 0b10 is assigned for 1000 Mbit ports (SPD of 0b11 is reserved for future use and should not be accessed). The default values for each of these registers are shown below. SPD 0b00 resets to 0x0258 (600 decimal) SPD 0b01 resets to 0x003C (60 decimal) SPD 0b10 resets to 0x0006 (6 decimal)

Table 135: Switch Management Register
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	Loopback Filter	RWR	Loopback filter. When this bit is cleared to a zero, normal operation occurs. When this bit is set to a one. Forward DSA frames that ingress a DSA port that came from the same Src_Dev will be filtered to the same Src_Port (i.e., the frame will not be allowed to egress the source port on the source device as indicated in the DSA Forward's Tag).
14	Reserved	RES	Reserved for future use.
13	Flow Control Message	RWR or RWS[1]	Enable Flow Control Messages[2]. When this bit is set to a one DSA Tag Flow Control messaged will be generated when a Flow Control enabled output queue becomes congested. When this bit is cleared to a zero DSA Tag Flow Control messages will not be generated but any received will be processed at the target MAC if flow control is enabled on the target MAC.
12	FloodBC	RWR	Flood Broadcast. When this bit is set to a one frames with the Broadcast destination address will flood out all the ports regardless of the setting of the port's Egress Floods bits (in Port Control, offset 0x04). VLAN rules and other switch policy still applies to these Broadcast frames. This bit only changes the policy of the Default Forward bit for Broadcast frames. When this bit is cleared to a zero frames with the Broadcast destination address are considered Multicast frames and will not egress out ports that have their Egress Flood bit cleared unless the Broadcast address is found in the address database.
11	Remove 1PTag	RWR	Remove One Provider Tag. When this bit is set to a one and a port is configured as a Provider Port (EgressMode = 0x3 in Port Control, port offset 0x04), recursive Provider Tag stripping will NOT be performed. Only the first Provider Tag found on the frame will be extracted and removed. Its extracted data will be used for switching. When this bit is cleared to a zero and a port is configured as a Provider Port (EgressMode = 0x3 in Port Control, port offset 0x04), recursive Provider Tag stripping will be performed. The first Provider Tag's data will be extracted and used for switching, and then all subsequent Provider Tags found in the frame will also be removed. This will only occur if the port's PortEType register (used to define the Provider Tag's EtherType) is not 0x8100 (can't perform recursive Provider Tag removal when the Provider's EtherType is equal to 0x8100).
10	ATUAge IntEn	RWS	ATU Age Violation Interrupt Enable. When a port is Locked (port offset 0x0B) an ATU Miss Violation will be generated when the frame's SA is not found in the address database. When this bit is set to a one an ATU Miss Violation will also be generated when a frame's SA is found in the address database but it has an Entry State value less than 0x4 (i.e., it is about half way aged out). RefreshLocked (port offset 0x0B) must not be enabled for this Age Violation to occur. Adding the ATU Age Violation to the ATU Miss Violation allows CPU directed learning to know an address is still being used before it ages out.

Table 135: Switch Management Register (Continued)
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
9	Tag Flow Control	RWR	Use and generate source port Flow Control status for Cross-Chip Flow Control. When this bit is set to a one bit 17 of the DSA Tag Forward frames is defined to be Src_FC and it is added to these frames when generated and it is inspected on these frames when received. When this bit is clearer to a zero bit 17 of the DSA Tag Forward frames is defined to be Reserved and it will be zero on these frames when generated and it will not be used on these frames when received (this is a backwards compatibility mode).
8	Reserved	RES	Reserved for future use.
7	ForceFlow ControlPri	RWS	Force Flow Control Priority. When this bit is set to a one the PRI[2:0] bits of generated DSAI Tag Flow Control frames will be set to the value of the FC Pri bits below. When this bit is cleared to a zero generated DSA Tag Flow Control frames will retain the PRI[2:0] bits from the frame that caused the congestion. This bit will have no effect if the FlowControlMessage bit (above) is cleared to a zero.
6:4	FC Pri	RWS to 0x7	Flow Control Priority. These bits are used as the PRI[2:0] bits on generated DSA Tag Flow Control frames if the ForceFlowControlPri bit above is set to a one.
3	Rsvd2CPU		Reserved multicast frames to CPU. This device supports two ways to support protocols that use multicast addresses. The first way is to enter the multicast address into the address database with a MGMT Entry_State, mapping it toward the CPU's port (Table 110). This allows proprietary protocols to be supported while also supporting standard protocols. If multiple address databases are used each multicast address will need to be added to the database for each database. The second way is to set this bit to a one. When this bit is a one frames with a Destination Address in the range 01:80:C2:00:00:0x or 01:80:C2:00:00:2x, regardless of their VLAN membership, will be considered MGMT frames and sent to the port's CPUDestPort (global offset 0x1A) as long as the associated Rsvd2Cpu Enable bit for the frame's DA is also set to a one (Global 2 offset 0x02 and 0x03). The MGMT Pri field (below) is used as the priority on these frames.
2:0	MGMT Pri	RWS to 0x7	MGMT Priority. These bits are used as the priority to use on Rsvd2CPU frames (above).

1. The FlowControlMessage bit will set to a one (enabled) if the SW_24P configuration pin is high, the HD_FLOW_DIS configuration pin in high and the FD_FLOW_DIS configuration pin is low at the rising edge of RESETn. This combination of configuration pins enables Cross-chip Flow Control on all Network ports when these ports are in either full or half-duplex mode of operation.
2. Flow Control Messages will egress out DSA links only, when the frame received on this link caused congestion. When Flow Control Messages are used the DSA link must have Flow Control enabled or some frame loss will occur.

Table 136: Device Mapping Table Register
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description
15	Update	SC	Update Target Device Routing data. When this bit is set to a one the data written to bits 3:0 will be loaded into the Target Device entry selected by the Trg_DevValue bits below. After the write has taken place this bit self clears to zero.
14:13	Reserved	RES	Reserved for future use.
12:8	Trg_Dev Value	RWR	Target Device Value. These bits select one of 32 possible Target Device Port register for both read and write operations to the Mapping Table. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
7:4	Reserved	RES	Reserved for future use.
3:0	Trg_Dev Port	RWS	Target Device Port number. These bits point to the physical port on this device where From_CPU frames will be routed by using the frame's Trg_Dev as an index into this table (when the Cascade Port, Global Control 2, Offset 0x1C, is set to a value of 0xF). In this way a physical mapping, or Routing Table, of the interconnection of the devices that make up the switch box or boxes in a stack is defined. When a write occurs to this register with the Update bit being a one these bits are written to the Trg_Dev Port selected by the Trg_Dev Value bits. When a write occurs to this register with the Update bit being a zero these bits are not written anywhere (this allow the Trg_Dev Value bits to be written to for read operations). When a read occurs to this register these bits reflect the Target Device Port data found for the entry selected by the Trg_Dev Value bits. The Routing Table is reset to the following values: Trg_Dev Value / Trg_Dev Port / Trg_Dev Value / Trg_Dev Port 0x00 / 0x0 / 0x10 / 0x0 0x01 / 0x1 / 0x11 / 0x1 0x02 / 0x2 / 0x12 / 0x2 0x03 / 0x3 / 0x13 / 0x3 0x04 / 0x4 / 0x14 / 0x4 0x05 / 0x5 / 0x15 / 0x5 0x06 / 0x6 / 0x16 / 0x6 0x07 / 0x7 / 0x17 / 0x7 0x08 / 0x8 / 0x18 / 0x8 0x09 / 0x9 / 0x19 / 0x9 0x0A / 0xA / 0x1A / 0xA 0x0B / 0xF / 0x1B / 0xF 0x0C / 0xF / 0x1C / 0xF 0x0D / 0xF / 0x1D / 0xF 0x0E / 0xF / 0x1E / 0xF 0x0F / 0xF / 0x1F / 0xF

Table 137: Trunk Mask Table Register
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15	Update	SC	Update Trunk Mask data. When this bit is set to a one the data written to bits 10:0 will be loaded into the Trunk Mask selected by the MaskNum bits below. After the write has taken place this bit self clears to zero.
14:12	MaskNum	RWR	Mask Number. These bits select one of eight possible Trunk Mask vectors for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
11	HashTrunk	RWR	Hash DA & SA for TrunkMask selection. Trunk load balancing is accomplished by using the frame's DA and SA fields to access one of eight Trunk Masks. When this bit is set to a one the hash computed for address table lookups is used for the TrunkMask selection. When this bit is cleared to a zero the lower 3 bits of the frame's DA and SA are XOR'ed together to select the TrunkMask to use.
10:0	TrunkMask	RWS	Trunk Mask bits. Bit 0 controls trunk masking for Port 0, bit 1 for Port 1, etc. When a write occurs to this register with the Update bit being a one these bits are written to the Trunk Mask selected by the MaskNum bits. When a write occurs to this register with the Update bit being a zero these bits are not written anywhere (this allow the MaskNum bits to be written to for read operations). When a read occurs to this register these bits reflect the Trunk Mask data found for the entry selected by the MaskNum bits. The TrunkMask is reset to all ones for all MaskNum entries.

Table 138: Trunk Members Table Register
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15	Update	SC	Update Trunk Routing data. When this bit is set to a one the data written to bits 10:0 will be loaded into the Trunk Route selected by the Trunk ID bits below. After the write has taken place this bit self clears to zero.
14:11	Trunk ID	RWR	Trunk Identifier. These bits select one of sixteen possible Trunk ID routing vectors for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
10:0	Trunk Members	RWR	Trunk Member bits. Bit 0 controls trunk routing for port 0, bit 1 for port 1, etc. When a write occurs to this register with the Update bit being a one these bits are written to the Trunk Member selected by the Trunk ID bits. When a write occurs to this register with the Update bit being a zero these bits are not written anywhere (this allow the Trunk ID bits to be written to for read operations). When a read occurs to this register these bits reflect the Trunk Member data found for the entry selected by the Trunk ID bits.

Table 139: Ingress Rate Command Register
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15	IRLBusy	SC	Ingress Rate Limit unit Busy. This bit must be set to a one to start an IRL operation (see IRLOp below). Only one IRL operation can be executing at one time so this bit must be zero before setting it to a one. When the requested IRL operation completes this bit will automatically be cleared to a zero.
14:12	IRLOp	RWR	Ingress Rate Limit unit Opcode. The devices support the following IRL operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Init all resources to the initial state 010 = Init the selected resource (pointed to by IRLPort and IRlRes) to the initial state. This initializes internal rate limiting related counters. 011 = Write to the selected resource/register (IRLUnit/IRLReg) 100 = Read the selected resource/register (IRLUnit/IRLReg) 101 = Reserved 110 = Reserved 111 = Reserved
11:8	IRLPort	RWR	Ingress rate limiting port. These bits indicate the ingress rate limiting port that is being accessed. Since there are 11 ports in the devices, these bits indicate one of the eleven ports. For example, if this field is programmed to a 0x3, it indicates that ingress rate resource belonging to port number 3 is being accessed.
7:5 (bits 7:6 are Reserved for the 88E6096 device)	IRLRes	RWR	Ingress rate limit resource. These bits indicate the ingress rate limit resource number that is being accessed. Since there are five rate limiting resources per port (two on the 88E6096 device), these bits indicate one of the five resources. For example, if this field is programmed to 0x2, it indicates that ingress rate resource 2 is being accessed.
4	Reserved	RWR	Reserved for future use.
3:0	IRLReg	RWR	Ingress Rate Limit register. These bits are used to define the controlling register being written or read on the resource defined in IRLUnit above. Use a value of 0x0 to access register 0, a value of 0x1 to access register 1, etc.

Table 140: Ingress Rate Data Register
Offset: 0x0A or Decimal 10

Bits	Field	Type	Description
15:0	IRLData	RWR	Ingress Rate Limit Data.

Table 141: Cross-chip Port VLAN Register
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
15	PVTBusy	SC	Port VLAN Table Busy. This bit must be set to a one to start a PVT operation (see PVTOp below). Only one PVT operation can be executing at one time so this bit must be zero before setting it to a one. When the requested PVT operation completes this bit will automatically be cleared to a zero.
14:12	PVTOp	RWR	Port VLAN Table Opcode. The device supports the following PVT operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Init the PVT Table to all one's (initial state) 010 = Reserved 011 = Write PVLAN Data (global 2, offset 0x0C) to the selected register[1] 100 = Read the selected register[2] 101 = Reserved 110 = Reserved 111 = Reserved
11:9	Reserved	RES	Reserved for future use.
8:0	Pointer	RWR	Pointer to the desired entry of the Cross-chip Port VLAN Table. These bits select one of 512 possible table entries for both read and write operations (defined by the PVTOp bits above). The meaning of the data bits in the table is described in the Cross-chip Port VLAN Data register below (global 2 offset 0x0C).

1. The register that gets written is the one pointed to by the Pointer register bits (bit 8:0 of this register)
2. The register that gets read is the one pointed to by the Pointer register bits (bit 8:0 of this register)

Table 142: Cross-chip Port VLAN Data Register
Offset: 0x0C or Decimal 12

Bits	Field	Type	Description
15:11	Reserved	RES	Reserved for future use.
10:0	PVLAN Data	RWS	Cross-chip Port VLAN Data used as a bit mask to limit where Cross-chip frames can egress (In-chip Port VLANs are masked using the VLANTable - port offset 0x06). Cross-chip frames are Forward frames that ingress a DSA or Ether Type DSA port (see Frame Mode in port offset 0x04)[1]. Bit 0 is a mask for port 0, bit 1 for port 1, etc. When a port's mask bit is one frames are allowed to egress that port on this device. When a port's mask bit is zero frames are not allowed to egress that port on this device. The entries in the Cross-chip Port VLAN Table are read and loaded by a CPU with the Cross-chip Port VLAN Addr register above (global 2 offset 0x0B). The 512 entry Cross-chip Port VLAN Table is accessed by ingressing frames based upon the original source port of the frame using the Forward frame's DSA tag fields Src_Dev, Src_Port/Src_Trunk and Src_Is_Trunk. The entry that is accessed by the frame is: If 5 Bit Port (in Global 2, offset 0x1D) = 0: If Src_Is_Trunk = 0 -> Src_Dev[4:0], Src_Port[3:0][2] If Src_Is_Trunk = 1 -> 0x1F, Src_Trunk[3:0] (i.e., at Src_Dev 0x1F) If 5 Bit Port (in Global 2, offset 0x1D) = 1: If Src_Is_Trunk = 0 -> Src_Dev[3:0], Src_Port[4:0][3] If Src_Is_Trunk = 1 -> 0xF, Src_Trunk[4:0] (i.e., at Src_Dev 0x0F) Cross-chip port VLANs with Trunks are supported in the table where this device's entries would be stored (defined by this device's Device Number). This portion of the table is available for Trunk entries because this device's port VLAN mappings to ports inside this device are masked by the port's VLANTable (port offset 0x06).

1. Cross-chip port VLANs cannot be supported on Ether Type DSA ports on Forward frames that don't contain a DSA Tag (Non-Forward DSA frames are not filtered by this table).
2. Only the lower 4 bits of the Src_Port are needed when interconnecting 88E6xxx switch devices since they all support less than 16 physical ports.
3. The full 5 bits of the Src_Port are needed when interconnecting this device with 98DXxxx switch devices since they support more than 16 physical ports. Only 16 Devices are supported in this mode, however.

Table 143: Switch MAC Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 will be loaded into the Swtich MAC octet register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:12	Reserved	RES	Reserved for future use.
11:8	Pointer	RWR	Pointer to the desired octet of Switch MAC. These bits select one of six possible Switch MAC registers for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
7:0	Data	RWR	Octet Data of the Switch MAC Address used as the switch's source address (SA) in transmitted full-duplex Pause frames. Six Switch MAC Octet data registers are accessed by using the Pointer bits above as follows: 0x0[7:1] = MAC Address Byte 0 (bits 47:41. Since bit 0 of byte 0 (bit 40) is the multicast bit (it is the 1st bit down the wire) it is always transmitted as a zero, and its value cannot be changed. 0x0[0] =DiffAddr. Different MAC Addresses per Port. This bit is used to have all ports transmit the same or different source addresses in full-duplex Pause frames. When this bits = 0, all ports transmit the same SA. When this bit = 1, each port uses a different SA where bits 47:4 of the MAC address are the same, but bits 3:0 are the port number (Port 0 = 0, Port 1 = 1, etc.) 0x1[7:0] = MAC Address Byte 1 (bits 39:32) used as the switch's source address (SA) in transmitted full-duplex Pause frames. 0x2[7:0] = MAC Address Byte 2 (bits 31:24) used as the switch's source address (SA) in transmitted full-duplex Pause frames. 0x3[7:0] = MAC Address Byte 3 (bits 23:16) used as the switch's source address (SA) in transmitted full-duplex Pause frames. 0x4[7:0] = MAC Address Byte 4 (bits 15:8) used as the switch's source address (SA) in transmitted full-duplex Pause frames. 0x5[7:0] = MAC Address Byte 5 (bits 7:0) used as the switch's source address (SA) in transmitted full-duplex Pause frames. Note: Bits 3:0 of this register are ignored if DiffAddr, above, is set to a one.

Table 144: ATU Stats Register
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
15:14	Bin	RWR	Bin selector bits. These bits are used to access the 4 Bin counters for static or non-static entries readable in bits 10:0 below. A value of 0x0 will access the lowest, or 1st bin to fill counter. 0x1 will access bin 1 and 0x2 will access bin 2's counter. A value of 0x3 will access the lowest, or last to fill bin counter.
13:12	CountMode	RWR	Bin Counter Mode. These bits determine what ATU entries are counted in the four Bin counters so various information can be extracted as follows: 00 = Count all valid entries 01 = Count all valid non-static entries only 10 = Count all valid entries found in the defined FID only 11 = Count all valid non-static entries found in the defined FID only The defined FID is the FID used during the ATU GetNext operation. These bits must be set prior to the start of an ATU GetNext so the ActiveBinCtrs contain this selected data at the end of the ATU GetNext.
11:0	ActiveBin Ctr	RO	Active ATU Bin Entry Counter. When a ATU GetNext operation is started the four Bin counters are all cleared to zero. When the ATU GetNext completes these four counters can be read and added together to get a total number of active MAC addresses that were currently found in the address data base using the CountMode above. Bin 0 is 1st bin to be used. Bin 1 is used when a Hash collision occurs and Bin 0 is already used. Bin 2 is used only after both bin 0 and 1 are filled, etc.

Table 145: Priority Override Table
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 3:0 will be loaded into the QPri Override register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:12	Reserved	RES	Reserved for future use.
11:8	Pointer	RWT	Pointer to the desired entry of the QPri Override table. These bits select one of sixteen possible QPri Override registers for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
7:4	Reserved	RES	Reserved for future use.
3	QPriEn	RWR	When this entry's bit is set to a one the Data bits below are used to overrided the frame's QPri. If this bit is cleared to a zero no QPri override will occur for this entry. Sixteen QPri override entries exist in the table. What each entry is used for is defined in the Data bits below.
2	Reserved	RES	Reserved for future use.

Table 145: Priority Override Table (Continued)
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
1:0	Data	RWR	Queue Priority Override Data. A value of 0x3 places a frame in the highest priority egress queue. A value of 0x0 places a frame in the lowest priority egress queue. When a frame enters a port its Type is determined (in priority order[1] if it could be multiple Types) and the frame's Type is used to access this table. If the Type's QPriEn bit (bit 3 above) is set to a one then the frame's QPri will be overridden with the value found in this Data field[2]. The frame Types supported are: 0x0 = Used on multicast DSA To_CPU frames with a Code of 0x0 (BPDU/MGMT). Not used on non-DSA Control[3] frames. 0x1 = Used on DSA To_CPU frames with a Code of 0x1 (Frame to Register Reply). Not used on non-DSA Control frames. 0x2 = Used on DSA To_CPU frames with a Code of 0x2 (IGMP/MLD Trap) and on non-DSA Control frames that are IGMP or MLD trapped (port offset 0x04). 0x3 = Used on DSA To_CPU frames with a Code of 0x3 (Policy Trap) and on non-DSA Control frames that are Policy Trapped (port offset 0x0E). 0x4 = Used on DSA To_CPU frames with a Code of 0x4 (ARP Mirror) and on non-DSA Control frames that are ARP Mirrored (port offset 0x08). 0x5 = Used on DSA To_CPU frames with a Code of 0x5 (Policy Mirror) and on non-DSA Control frames that are Policy Mirrored (port offset 0x0E). 0x6 = Used on DSA To_CPU frames with a Code of 0x6 (Reserved). Not used on non-DSA Control frames. 0x7 = Used on unicast DSA To_CPU frames with a Code of 0x0 (unicast MGMT). Not used on non-DSA Control frames. 0x8 = Used on DSA From_CPU frames. Not used on non-DSA Control frames. 0x9 = Used on DSA Cross-chip Flow Control frames. Not used on non-DSA Control frames. 0xA = Used on DSA Cross-chip Egress Monitor frames. Not used on non-DSA Control frames.

Table 145: Priority Override Table (Continued)
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
1:0 (cont.)	Data (cont.)	RWR	0xB = Used on DSA Cross-chip Ingress Monitor frames. Not used on non-DSA Control frames. 0xC = Used on normal network ports (FrameMode = 0x0, port offset 0x04) on frames whose Ethertype matches the port's PortEType register. Not used on DSA Control frames. 0xD = Used on Non-DSA Control frames that contain a Broadcast destination address. Not used on DSA Control frames. 0xE & 0xF = Reserved

1. Priority order (low to high): Broadcast, PolMirror, PolTrap, ETYPE, ARP, IGMP/MLD, MGMT.
2. If a frame can map to more that one item (like an ARP can also be a Broadcast) the last one on the list will try to be used (ARP in the example) even if that entry is not enabled in the table and the previous decode was (e.g., ARP was not enabled but Broadcast was, the ARP frame will NOT get priority overridden).
3. Non-DSA Control frames are all frames that enter a non-DSA port and DSA Forward frames.

Table 146: SMI PHY Command Register[1]
Offset: 0x18 or Decimal 24

Bits	Field	Type	Description
15	SMIBusy	SC	SMI PHY Unit Busy. This bit must be set to a one to start an internal SMI operation on the SMI_PHY pins (see SMIOp below). Only one SMI operation can be executing at one time so this bit must be zero before setting it to a one. When the requested SMI operation completes this bit will automatically be cleared to a zero. If the PPU is disabled this bit clears right away.
14:13	Reserved	RES	Reserved for future use.
12	SMIMode	RWR	SMI PHY Mode bit. This bit is used to define the SMI frame type to generate as follows: 0 = Generate IEEE 802.3 Clause 45 SMI frames 1 = Generate IEEE 802.3 Clause 22 SMI frames
11:10	SMIOp	RWR	SMI PHY Opcode. These bits are used to select the SMI opcode to operate on during SMI commands as follows: When the SMIMode bit = 1 then SMIOp = (IEEE 802.3 Clause 22): 00 = Reserved 01 = Write Data Register 10 = Read Data Register 11 = Reserved When the SMIMode bit = 0 then SMIOp = (IEEE 802.3 Clause 45): 00 = Write Address Register 01 = Write Data Register 10 = Read Data Register 11 = Read Data Register with post increment on the Address Register
9:5	DevAddr	RWR	SMI PHY Device Address bits. These bits are used to select the SMI device (Clause 22) or port (Clause 45) to operate on during SMI commands.
4:0	RegAddr	RWR	SMI PHY Register Address bits. These bits are used to select the SMI register (Clause 22) or device class (Clause 45) to operate on during SMI commands.

1. This register can be used to access the PHY registers only when the PPU is enabled (global offset 0x04).

Table 147: SMI PHY Data Register[1]
Offset: 0x19 or Decimal 25

Bits	Field	Type	Description
15:0	SMIData	RWR	SMI PHY Data register. During SMI Writes these bits must be written with the SMI PHY data to be written prior to starting the SMI PHY operation (i.e., before setting SMIBusy to a one). During SMI PHY Reads these bits will contain the SMI PHY data that was read after the SMI PHY read operation completes (i.e., SMIBusy returns to a zero). Writes to this register must not be done while SMIBusy is a one. If the PPU is disabled this data will be 0xFFFF.

1. This register can be used to access the PHY registers only when the PPU is enabled (global offset 0x04).

Table 148: Watch Dog Control Register
Offset: 0x1B or Decimal 27

Bits	Field	Type	Description
15:8	Reserved	RES	Reserved for future use.
7	EgressWD Event	RO	Egress Watch Dog Event. If any port's egress logic detects an egress watch dog issue, this bit will be set to a one, regardless of the setting of the Egress Watch Dog bit below. This bit will be cleared by a SWReset (Global offset 0x04).
6	RMU TimeOut	RWS	Remote Management TimeOut. When this bit is set to a one the Remote Management Unit (RMU) will timeout on Wait on Bit commands. If the bit that is being tested has not gone to the specified value after 1 second has elapsed the Wait on Bit command will be terminated and the Response frame will be sent without any further processing. When this bit is cleared to a zero the Wait on Bit command will wait until the bit that is being tested has changed to the specified value.
5	QC Watch Dog	RWS	Queue Controller Watch Dog Enable. When this bit is set to a one the QC's watch dog circuit checks for link list errors and any errors found in the QC show up in the Watch Dog History bit below. When this bit is cleared to a zero, QC watch dog events will be ignored.
4	Egress WD History	RO	Egress Watch Dog History. If any port's egress logic detects an egress watch dog issue, this bit will be set to a one, regardless of the setting of the Egress Watch Dog bit below. This bit can only be cleared by a hardware reset.
3	Egress Watch Dog	RWS	Egress Watch Dog Enable. When this bit is set to a one each port's egress circuit checks for problems between the port and the Queue Controller. Any errors found will show up in the Watch Dog History bit below. When this bit is cleared to a zero, Egress watch dog events will be ignored with the exception that the Egress WD History bit, above, will still be set on any egress watch dog event.
2	Force Watch Dog	RWR	Force a Watch Dog Event. When this bit is set to a one a watch dog event is forced as if an enabled watch dog event occurred. This bit allows the testing of software that is designed to service the watch dog events. This bit is cleared by a SW Reset so it will automatically be cleared to zero if the SWReset on WD bit below is set to a one.

Table 148: Watch Dog Control Register (Continued)
Offset: 0x1B or Decimal 27

Bits	Field	Type	Description
1	Watch Dog History	RO	Watch Dog History. When this bit is set to a one, some enabled Watch Dog event occurred. Three events are possible and they are: • A QC Watch Dog event occurred when the QC Watch Dog (enable bit, above) was set to a one. • An Egress Watch Dog event occurred when the Egress Watch Dog (enable bit, above) was set to a one. • A Watch Dog event was forced by setting the Force Watch Dog (bit, above) to a one. This bit can only be cleared by a hardware reset.
0	SWReset on WD	RWS	SWReset on Watch Dog Event. When this bit is set to a one, any enabled watch dog event will automatically reset the switch core's data path just as if the SWReset bit (global 1, offset 0x04) was set to a one. The Watch Dog History (bit, above) nor the Watch Dog Int (global 2, offset 0x00) will not be cleared by this automatic SWReset. This allows the user to know if any watch dog event ever occurred even if the switch is configured to automatically recover from a watch dog. When this bit is cleared to a zero enabled watch dog events will not cause a SWReset. The Watch Dog History (bit, above) and the Watch Dog Int (global 2, offset 0x00) will still be set on enabled watch dog events, however.

Note

The QoS Weights Register is only available for the 88E6097F and 88E6097 devices. These registers are reserved in the 88E6096 device.

Table 149: QoS Weights Register
Offset: 0x1C or Decimal 28

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 will be loaded into the QoS Weights octet register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14	Reserved	RES	Reserved for future use.
13:8	Pointer	RWR	Pointer to desired octet of QoS Weights. These bits select one of 32 possible QoS Weight Data registers and the QoS Weight Length register for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
7:0	Data	RWS to see text	Octet Data of the programmable QoS Weight table. 33 QoS Weight registers are accessed by using the Pointer bits above as follows: 0x00 to 0x1F = QoS Weight Table Data. 0x20 = QoS Weight Table Length. The QoS Weight Table Data is a 128 x 2 bit table where each Weight Table Data entry contains four 2-bit entries. Bits 1:0 of the entry at Pointer 0x00 is the 1st table entry. Bits 3:2 are the 2nd entry, etc. The 5th entry is bits 1:0 at Pointer 0x01. The two-bit wide entries are used to contain the desired queue processing priority order (starting with bits 1:0 at Pointer 0x00). The QoS Weight Table Length register is used to define the length of the QoS Weight Table Data. Writing to this register causes the new table to be used by the Queue Controller (so the data at pointers 0x00 to 0x1F must be written 1st). The hardware reset values of this table default to an 8, 4, 2, 1 weight as follows: 0x00 = 0x7B (this defines a 3, 2, 3, 1 order) 0x01 = 0x3B (this defines a 3, 2, 3, 0 order) 0x02 = 0x7B (this defines a 3, 2, 3, 1 order) 0x03 = 0x3B (this defines a 3, 2, 3 order) 0x04 to 0x1F = 0x00 0x20 = 0x0F (this indicates the 1st 15 steps in the table are to be used)

Table 150: SDET Polarity Register
Offset: 0x1D or Decimal 29

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	5 Bit Port	RWR	Use 5 bits for Port data in the Port VLAN Table (PVT). When this bit is set to a one the 9 bits used to access the PVT memory is: Addr[8:5] = Source Device[3:0] or Device Number[3:0][1] Addr[4:0] = Source Port/Trunk[4:0] When this bit is cleared to a zero the 9 bits used to access the PVT memory is: Addr[8:4] = Source Device[4:0] or Device Number[4:0] Addr[3:0] = Source Port/Trunk[3:0]
13	NoEgr Policy	RWR	No Egress Policy. When this bit is set to a one Egress 802.1Q Secure and Check discards are not performed. This mode allows a non-802.1Q enabled port to send a frame to an 802.1Q enabled port that is configured in the Secure or Check 802.1Q mode (see port offset 0x08). In this situation the frames will egress even if the VID assigned to the frame is not found in the VTU. When this bit is cleared to zero and the Egress port's 802.1Q mode is Secure or Check (see port offset 0x08) the VID assigned to all frames mapped to this port must be found in the VTU or the frame will not be allowed to egress this port.
12:11	Reserved	RES	Reserved for future use.
10:0	SDET Polarity	RWS	SDET (Signal Detect) Polarity select bits for each port. Bit 10 is for Port 10, bit 9 is for Port 9, etc. SDET is used to help determine link on fiber ports. This bit affects the active level of a port's SDET pins as follows: 0 = SDET is active low. A low level on the port's SDET pin is required for link to occur. 1 = SDET is active high. A high level on the port's SDET pin is required for link to occur. SDET is used when the port is configured as a fiber port. In all other port modes the SDET pins are ignored and these bits have no effect.

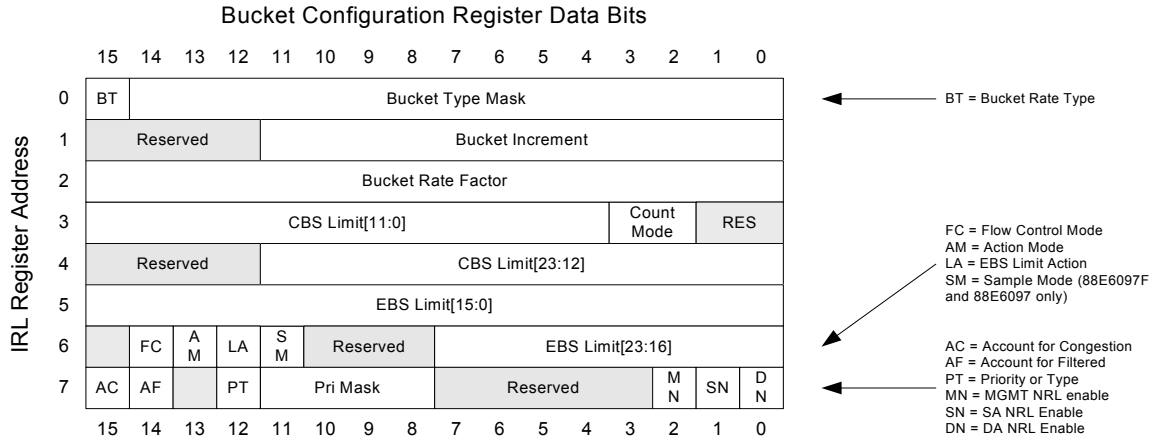
1. Source Device is used if the frame's Src_Is_Trunk = 0, else this device's Device Number is used.

13.6.1 PIRL Registers

The devices contain a set of Port Ingress Rate Limiting (PIRL) registers that effect the selected Ethernet ports in the device. Each PIRL register is 16-bits wide and their bit assignments are shown in Figure 76.

There are five sets of these registers per port, one set per PIRL resource or bucket. These registers are accessible by Global 2 registers at offsets 0x09 and 0x0A (Ingress Rate Command register and Ingress Rate Data register).

Figure 76: PIRL Register bit Map (from Global 2 offsets 0x09 & 0x0A)



A detailed description of the PIRL registers follow.

Table 151: PIRL Bucket Configuration Register
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15	BktRate Type	RWR	Bucket Rate Type 0 = Indicates the bucket is traffic type based. 1 = Indicates the bucket is rate based. **NOTE:** When a rate resource is configured to be rate based, then neither PriMask setting nor BktTypeMask setting have any affect on the rate resource.
14:0	BktType Mask	RWR	This field has the following definition if BktRateType = 1'b0; [0] – Unknown Unicast The definition of unknown unicast in the context of PIRL is that if the MAC DA search resulted in a failure and if the ingress parsing engine does not classify the frame as either PolMirror or PolTrap. If the ingress parsing engine did mark a frame as an unknown and either PolMirror or PolTrap, then such packets would be tracked based on bits 13 and 14 of BktTypeMask field. [1] – Unknown Multicast The definition of unknown multicast in the context of PIRL is that if the MAC DA search resulted in a failure and if the ingress parsing engine does not classify the frame as either PolMirror or PolTrap. If the ingress parsing engine did mark a frame as an unknown and either PolMirror or PolTrap, then such packets would be tracked based on bits 13 and 14 of BktTypeMask field. [2] – Broadcast [3] – Multicast [4] – Unicast [5] – MGMT Frames [6] – Reserved [7] – ARP [8] – TCP Data [9] – TCP Ctrl (if any of the TCP FLAGS[5:0] bits are set) [10] – UDP [11] – NON_TCPUDP (covers IGMP, ICMP, GRE, IGRP, L2TP) [12] – IMS (Ingress Monitor Source) [13] – PolicyMirror (88E6097/88E6097F only) [14] – PolicyTrap (88E6097/88E6097F only)

Table 152: PIRL Bucket Configuration Register
Offset: 0x1 or Decimal 1

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
11:0	Bkt Increment	RWR	Bucket increment. This parameter indicates the amount of tokens that need to be added for each byte of frame information.

Table 153: PIRL Bucket Configuration Register
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
15:0	BktRate Factor	RWR	Bucket Rate Factor. This is a factor which determines the amount of tokens that need to be decremented for each rate resource decrement (which is done periodically based on the Committed Information Rate). The higher the value of this field the higher will be the decrement value. If SMode is programmed to a one, then BktRateFactor is expected to be programmed to a zero.

Table 154: PIRL Bucket Configuration Register
Offset: 0x3 or Decimal 3

Bits	Field	Type	Description
15:4	CBSLimit [11:0]	RWR	Committed Burst Size limit (lower 12 bits). This indicates the committed information burst amount. The upper bits of this register is at IRL offset 0x4 (the next register below).
3:2	Count Mode	RWR	Frame bytes to be accounted for in the rate resource's rate limiting calculations. The supported configurations of this field (bits 1 down to 0) are: 00 = Frame based 01 = Count all Layer1 bytes 10 = Count all Layer2 bytes 11 = Count all Layer3 bytes Frame based configures the rate limiting resource to account for the number of frames from a given port mapped to this rate resource. Layer1 = Preamble (8 Bytes) + Frame's DA to CRC + IFG (12 bytes) Layer2 = Frame's DA to CRC Layer3 = Frame's DA to CRC – 18[1] – 4 (if frame is tagged[2]) A frame is considered tagged if it is either Customer or Provider tagged during ingress.
1:0	Reserved	RES	Reserved for future use.

1. The 18 bytes are: 6 for DA, 6 for SA, 2 for EtherType and 4 for CRC.
2. Only one tag is counted even if the frame contains more than one tag (i.e., it is Provider tagged).

Table 155: PIRL Bucket Configuration Register
Offset: 0x4 or Decimal 4

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:0	CBSLimit [23:12]	RWR	Committed Burst Size limit (upper 12 bits). This indicates the committed information burst amount. The lower bits of this register is at IRL offset 0x3 (the next register above).

Table 156: PIRL Bucket Configuration Register
Offset: 0x5 or Decimal 5

Bits	Field	Type	Description
15:0	EBSLimit	RWR	Excess Burst Size limit (lower 16 bits). The upper bits of this register is at IRL offset 0x6 (the next register below). If the ingress rate resources' BktTokenCount exceeds the EBSLimit, based on the equation {EBSLimit – BktTokenCount < CBSLimit} EBSLimitAction is taken on the incoming frame. Note that if ActionMode=1, the BktTokenCount can exceed EBSLimit and if the ActionMode = 0, the BktTokenCount is clamped at EBSLimit.

Table 157: PIRL Bucket Configuration Register
Offset: 0x6 or Decimal 6

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	PirlFC Mode	RWR	Port Ingress Rate Limit Flow Control Mode. This bit is used to determine when the flow control (asserted due to ingress rate limiting threshold exceeding reasons) gets deasserted. 0 = De-assert flow control when ingress rate resource has become empty i.e., the ingress rate resource can accept more frames as it is empty. 1 = De-assert flow control when ingress rate resource has enough room to accept at least one frame of size determined by the value programmed in the CBS_Limit field as specified in PIRL offset 0x3 & 0x4. For example, if the CBS_Limit for the ingress rate resource is programmed to be 2k Bytes, then the flow control will get de-asserted if there is at least 2k Bytes worth of tokens available in the ingress rate resource.
13	Action Mode[1]	RWR	PIRL Action mode. When enabled (by setting this bit to one) configures this rate limiting resource to accept an incoming frame even though there are not enough tokens to accept the entire incoming frame. When disabled (by clearing this bit to zero) configures this rate limiting resource to take an action specified in the EBSLimitAction if there is not enough tokens available to accept the entire incoming frame.
12	EBSLimit Action	RWR	If the incoming port information rate exceeds the EBS_Limit, this field specifies the action that needs to be taken for the violating traffic. 0 = indicates that the frame that was received on the port that this particular rate resource is assigned to will get discarded if the EBSLimit has been exceeded. 1 = indicates that a flow control frame could get sent back to the source if the flow control is enabled for that port and if EBSLimit has been exceeded. Note: If any of the PIRL resource buckets for this port were to discard the packet the packet would get discarded by the switch and similarly if any of the PIRL resource buckets for this port were to send a flow control packet back to the source the flow control packets does get sent.

Table 157: PIRL Bucket Configuration Register (Continued)
Offset: 0x6 or Decimal 6

Bits	Field	Type	Description
11	SMode (88E6097F and 88E6097 devices only - reserved for the 886096 device)	RWR	Sampling Mode. This mode is used for sampling one out of so many frames / bytes (determined by the configured rate resource parameters) for a stream of frames (determined by the BktTypeMask configuration) that are being monitored. The stream could be identified by the ingress engine as a Policy mirror and packet sampling can be applied for that stream using one of the rate resources. In this mode, once the rate resource's EBSLimit is exceeded, the next incoming frame from this port that is assigned to this resource gets sent out to the mirror destination. After sending a sample frame, the token count within the rate resource is reset to zero and the bucket increments continue for each subsequent frame arrival. When this bit is set to a one, the sampling mode is enabled and by clearing this bit to a zero the sampling mode is disabled. The sampling mode is useful for limiting the number of Mirror frames sent to the mirror destination or for sampling any of the frame types defined in the BktTypeMask field (RateType = 0) or for sampling frames from a given port (if RateType is programmed to 1). In the 88E6097, since there are five rate resources per port and given that each of these rate resources can be programmed to track different types of traffic streams with different bucket limits, if any of the bucket's logic were to decide that the frame needs to be discarded then the frame would not get Sampled to the mirror destination.
10	Reserved	RES	Reserved for future use.
7:0	EBSLimit	RWR	Excess Burst Size limit (upper 8 bits). The lower bits of this register is at IRL offset 0x5 (the next register above). If the ingress rate resources' BktTokenCount exceeds the EBSLimit, based on the equation {EBSLimit – BktTokenCount < CBSLimit} EBSLimitAction is taken on the incoming frame. **NOTE:** If ActionMode=1, the BktTokenCount can exceed EBSLimit and if the ActionMode = 0, the BktTokenCount is clamped at EBSLimit.

1. This bit is expected to be enabled for TCP based applications and disabled for media streaming kind of applications where timing of the data is critical.

Table 158: PIRL Bucket Configuration Register
Offset: 0x7 or Decimal 7

Bits	Field	Type	Description
15	AcctForQCong	RWR	Account for queue congestion discards. When enabled (by setting this bit to one), this bit indicates that even if the incoming frames are discarded in the chip because of output port queue congestion, the ingress rate limiting logic accounts for the incoming bytes (or frames if CountMode is configured to be 0x0) in the bucket_increment calculations. When disabled (by clearing this bit to zero), this bit indicates that if the incoming frames are discarded in the chip because of output port queue congestion, the ingress rate limiting logic does not account for the incoming bytes (or frames if CountMode is configured to be 0x0) in the bucket_increment calculations.
14	AcctForFilt	RWR	Account for filtered frames. When enabled (by setting this bit to one), this bit indicates to the rate limiting logic that even if the incoming frames are filtered in the chip because of ingress policy reasons, account for the incoming bytes (or frames if the CountMode is configured to be 0x0) in the bucket_increment calculations. If SMode = 1, AcctForFilt enabling would allow the ingress policy filtered frames which are classified by the ingress block as PolMirror to the mirror destination. When disabled (by clearing this bit to zero), this bit indicates to the rate limiting logic that if the incoming frames are filtered in the chip because of ingress policy reasons, do not account for the incoming bytes (or frames if the CountMode is configured to be 0x0) in the bucket_increment calculations.
13	Reserved	RES	Reserved for future use.
12	PriOrPT	RWR	Priority Or Packet Type. When this bit is set to a one, the frame types defined in the BktTypeMask field (see below) OR the priority bits defined in PriMask (see below) field, determine the incoming frames that get rate limited using this ingress rate resource. When this bit is cleared to a zero, the frame types defined in the BktTypeMask field AND the priority bits defined in PriMask field determine the incoming frames that get rate limited using this rate resource. For example if BktTypeMask[4] = 1 (i.e., unicast frames) and PriMask[3:0] = 0x4 (priority 2 frames), if PriORPT is set to a one, then either unicasts or Priority 2 frames are accounted for in the ingress limiting calculations for this rate resource. If PriORPT is cleared to a zero, then all unicast frames that are classified to be priority2 frames are accounted for in the ingress limiting calculations for this rate resource.

Table 158: PIRL Bucket Configuration Register (Continued)
Offset: 0x7 or Decimal 7

Bits	Field	Type	Description
11:8	PriMask	RWR	Priority Mask. Each bit indicates one of the four queue priorities. Setting each one of these bits indicates that this particular rate resource is slated to account for incoming frames with the enabled bits' priority. For example, if bits zero and two are set i.e., this field is set to 0x5, frames with queue priority of zero and two are accounted for in this ingress rate resource. Note that if PriOrPT bit affects if all frames with a certain priority get rate limited using this rate resource or not (refer to the PriOrPT field description). If this field is set to 0x0, priority of the frame doesn't have any affect on the ingress rate limiting calculations done for this ingress rate resource.
7:3	Reserved	RES	Reserved for future use.
2	MGMT NrlEn	RWR	Management Non Rate Limit Enable. When this bit is cleared to a zero all frames that are classified by the ingress frame classifier as MGMT frames would be considered to be ingress rate limited as far as this particular ingress rate resource is concerned. When this bit is set to a one, all frames that are classified as MGMT frames by the ingress frame classifier would be excluded from the ingress rate limiting calculations for this particular ingress rate resource. In trusted[1] network environments, MGMT frames could be passing through the switches purely for network administration and management of the switches. In such scenarios it may be necessary to not consider MGMT frames as part of the end customers traffic. This bit provides an option to either consider or not consider MGMT frames as part of the ingress rate limiting for a given rate resource.
1	SANrlEn	RWR	SA Non Rate Limit enable. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one then SA ATU non rate limiting overrides can occur on this port. An SA ATU non rate limiting override occurs when the source address of a frame results in an ATU hit where the SA's MAC address returns an EntryState that indicates Non Rate Limited[2]. When this occurs the frame will not be ingress rate limited.
0	DANrlEN	RWR	DA Non Rate Limit enable. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one then DA ATU non rate limiting overrides can occur on this port. A DA ATU non rate limiting override occurs when the destination address of a frame results in an ATU hit where the DA's MAC address returns an EntryState that indicates Non Rate Limited. When this occurs the frame will not be ingress rate limited.

1. Trusted networks in this context are those which have customers connected to the switch that are never expected to generate DoS attacks using MGMT frames.
2. SA Non Rate Limiting Override can only occur on MAC addresses that are Static or where the Port is Locked, and where the port is the mapped source port for the MAC address.

Section 14. EEPROM Programming Format

The devices support an optional external serial EEPROM device for programming its internal registers. The EEPROM data will be read in once after Reset is deasserted unless the Stand Alone Switch Mode is selected (SW_MODE[1:0] = 0b10 – Table 17).

The devices support 1K bit (24C01), 2K bit (93C56 or 24C02) or 4K bit (93C66 or 24C04) EEPROM devices (the size of the device is selected by the EE_DIN/EE_1K pin at reset – Table 16). The external EEPROM device must be configured in x16 data organization mode.

No matter what device is attached, the EEPROM device is read and processed in the same way:

1. Start at EEPROM address 0x00.
2. Read in the 16-bits of data from the current address, this is called the Command.
3. If the just read in Command is all one's, terminate the serial EEPROM reading process, go to 8.
4. Increment the address by 1 (to the next address). If the Command does not need any data from the EEPROM, process the Command and go to step 2.
5. Read in the 16-bits of data from the next address, this is called RegData and increment the address by 1.
6. Write RegData to the location or locations defined by the previous Command.
7. Go to 2.
8. Set the EEInt bit in Global Status to a one (global 0x00) generating an Interrupt (if enabled).
9. Done.

The 16-bit Command determines which register or registers inside the devices are updated as follows (refer to Figure 76):

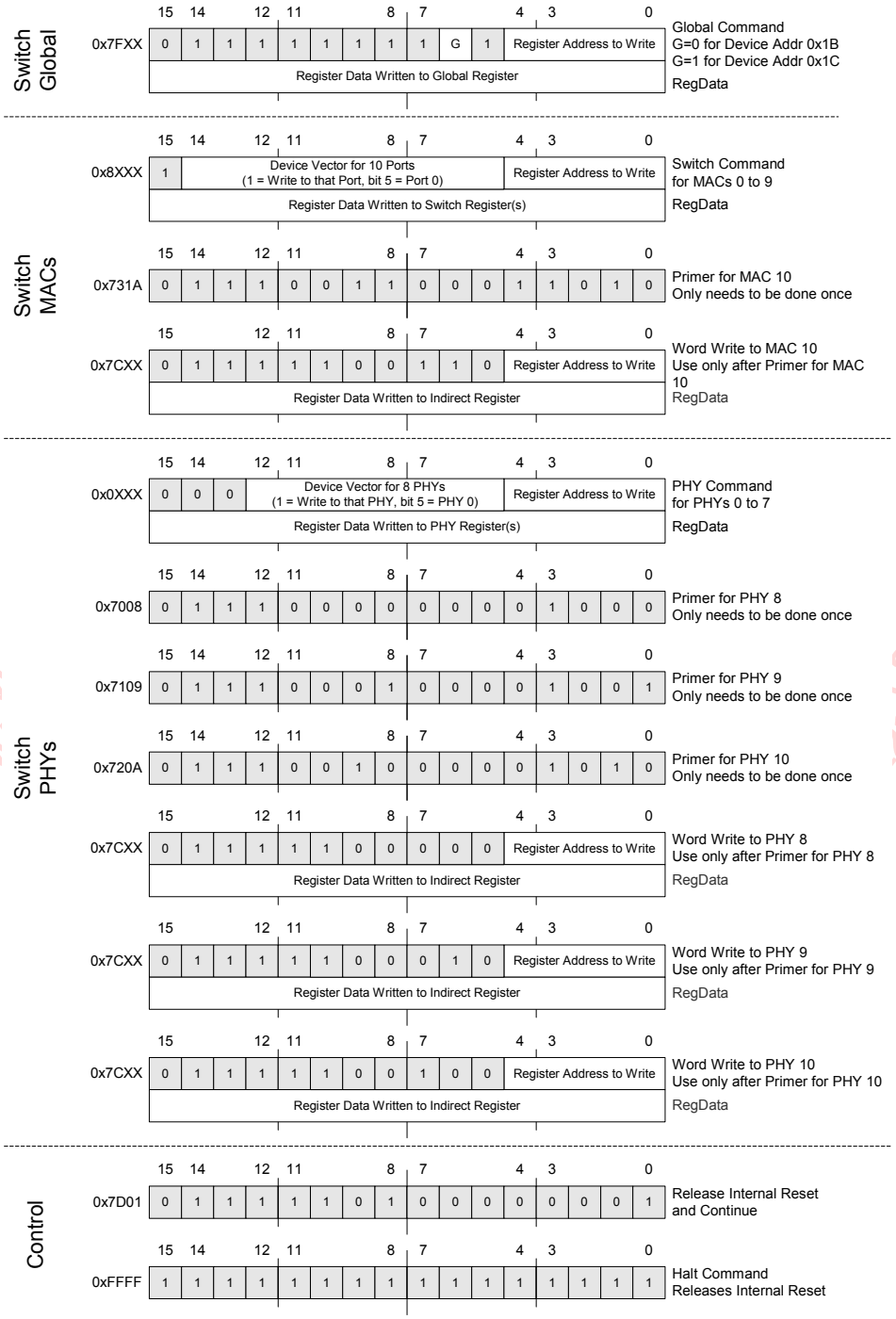
1. A Switch Global register is written to if the upper 8-bits of the Command is 0x7F. The Global space written to is determined by bit 6, the G bit. If G=0 Global 1 at Device Addr 0x1B is written to. If G=1 Global 2 at Device Addr 0x1C is written to. Bits 4 to 0 determine the register to load.
2. Bit 15=1 the lower ten Switch MAC registers are written to (SMI Device Addresses 0x10 to 0x19). A Device Vector (Command bits 14 to 5) is used to determine which port or ports MAC is to be written to. One Command can be used to write the same data to all 10 MACs or to just some of them. Bits 4 to 0 determine the register to load.
3. The device contains 11 ports so the previously described Command cannot be used to reach MAC 10. Instead, a separate command is available for this. If the upper 10 bits of the Command is 0x7CC then port 10's MAC will be written to if a primer command has been executed first. Bits 4 to 0 determine the register to load. The Primer for MAC 10 Command (0x731A) only needs to be executed once, but it must be executed before any Commands to write to port 10's MAC.
4. When the upper 3 bits of the Command is 0x0, the lower eight PHY registers are written to (SMI Device Addresses 0x00 to 0x07). A Device Vector (Command bits 12 to 5) is used to determine which port or ports PHY is to be written to. One Command can be used to write the same data to all 8 PHYs or to just some of them. Bits 4 to 0 determine the register to load. The PPU must be disable or the writes to the PHYs will not occur.
5. The device contains 11 ports so the previously described Command cannot be used to reach PHYs 8 to10. Instead, separate commands are available for this. If the upper 10 bits of the Command is 0x7C0 then port 8's PHY will be written to if a primer command has been executed first. Bits 4 to 0 determine the register to load. The Primer for PHY 8 Command (0x7008) only needs to be executed once, but it must be executed before any Commands to write to port 8's PHY. Port 9's PHY and port 10's PHY have separate Commands to write to the PHY (0x7C4x and 0x7C8x, respectively) and separate primer commands (0x7109 and 0x720A respectively). The PPU must be disable or the writes to the PHYs will not occur.

6. The EEPROM updates the internal register settings and then releases an internal Reset when it reads a Command of all 1's (a Halt Command). This allows register setting changes to be made prior to letting packet flow through the switch (required to prevent VLAN leaks). But the ATU and VTU cannot be loaded with data from the EEPROM unless the internal Reset is released. A Command of 0x7D01 will release the internal Reset and continue EEPROM processing so the ATU or VTU can be loaded from the EEPROM. This will also allow packets to start flowing through the switch so placement of this Command is important.

The Command/RegData list can be as short or as long as needed. This makes optimum use of the limited number of Command/RegData pairs that can fit in a given size EEPROM (31[1] Command/RegData pairs in the 1K bit device, 63 in the 2K and 127 in the 4K bit devices).

1. The maximum number of Command/RegData pairs is one less than expected because the last entry must be the End of List Indicator of all one's.

Figure 77: EEPROM Data Format



Section 15. PHY Register Description

The device contains eight physical layer devices (PHYs). These devices are accessible using SMI device addresses 0x00 to 0x07 depending upon the value of the P8_MODE, P9_MODE[2:0], and P10_MODE[2:0] pins at reset. The PHYs are fully IEEE 802.3 compliant including their register interface.

The PHYs in the devices are identical to those in the Marvell® 88E3082 Octal Transceiver.

Table 159: PHY Register Map

Description	Offset Hex	Offset Decimal	Page Number
PHY Control Register	0x00	0	page 350
PHY Status Register	0x01	1	page 353
PHY Identifier	0x02	2	page 355
PHY Identifier	0x03	3	page 355
Auto-Negotiation Advertisement Register	0x04	4	page 356
Link Partner Ability Register (Base Page)	0x05	5	page 358
Link Partner Ability Register (Next Page)	0x05	5	page 358
Auto-Negotiation Expansion Register	0x06	6	page 359
Next Page Transmit Register	0x07	7	page 360
Link Partner Next Page Register	0x08	8	page 361
Reserved Registers	0x09-0x0F	9 - 15	page 361
PHY Specific Control Register I	0x10	16	page 362
PHY Specific Status Register	0x11	17	page 365
PHY Interrupt Enable	0x12	18	page 367
PHY Interrupt Status	0x13	19	page 369
PHY Interrupt Port Summary	0x14	20	page 371
PHY Receive Error Counter	0x15	21	page 372
LED Parallel Select Register	0x16	22	page 373
LED Stream Select for Serial LEDs (Global Register) - 88E6097F Only	0x17	23	page 375
PHY LED Control Register	0x18	24	page 377
PHY Manual LED Override Register	0x19	25	page 378
VCT™ Control Register	0x1A	26	page 380
VCT™ Status Register	0x1B	27	page 381
PHY Specific Control Register II	0x1C	28	page 382
Reserved Registers	0x1D to 0x1F	29 - 31	page 382

Table 160: PHY Control Register
Offset: 0x00 (Hex), or 0 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	SWReset	R/W, SC	0x0	0	PHY Software Reset Writing a 1 to this bit causes the PHY state machines to be reset. When the reset operation is done, this bit is cleared to 0 automatically. The reset occurs immediately. 0 = Normal operation 1 = PHY reset
14	Loopback	R/W	0x0	Retain	Enable Loopback Mode When loopback mode is activated, the transmitter data presented on TXD is looped back to RXD internally. The PHY has to be in forced 10 or 100 Mbps mode. Auto-Negotiation must be disabled. 0 = Disable loopback 1 = Enable loopback
13	SpeedLSB	R/W	See Descr	Update	Speed Selection (LSB) The speed selection (MSB or LSB) determines the forced speed if Auto-Negotiation is disabled. If Auto-Negotiation is enabled, the speed ability advertisement is located in Register 4, and 0.13 and 0.6 have a "don't care" definition. A write to this register bit has no effect unless any one of the following also occurs: Software reset is asserted (bit 15) or Power down (bit 11) transitions from power down to normal operation. 0 = 10 Mbps 1 = 100 Mbps After hardware reset, the default value is as follows: 88E6096/88E6097 device = 1 88E6097F device = Determined by P[7:0]_CONFIG

Table 160: PHY Control Register (Continued)
Offset: 0x00 (Hex), or 0 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
12	AnegEn	R/W	See Descr	Update	Auto-Negotiation Enable If Auto-Negotiation is enabled (0.12 = 1), the speed and duplex ability advertisement are located in Register 4, and register bits 0.13, 0.6, and 0.8 have a "don't care" definition. A write to this register bit has no effect unless any one of the following also occurs: Software reset is asserted (bit 15, above), Power down (bit 11, below), or the PHY transitions from power down to normal operation. If Auto-Negotiation is already enabled, any link drop event will cause the Auto-Negotiation process to start again. 0 = Disable Auto-Negotiation Process 1 = Enable Auto-Negotiation Process After hardware reset the default value is as follows: 88E6096/88E6097 device 0.12 = 1 88E6097F device = Determined by P[7:0]_CONFIG
11	PwrDwn	R/W	0x0	Retain	Power Down Mode When the port is switched from power down to normal operation, software reset and restart Auto-Negotiation are performed even when bits Reset (bit 15, above) and Restart Auto-Negotiation (bit 9, below) are not set by the user. 0 = Normal operation 1 = Power down
10	Isolate	RO	Always 0	Always 0	Isolate Mode Will always be 0. The Isolate function is not available, since full MII is not implemented. 0 = Normal operation
9	RestartAneg	R/W, SC	0x0	Self Clear	Restart Auto-Negotiation Auto-Negotiation automatically restarts after hardware or software reset regardless of whether or not the restart bit is set. 0 = Normal operation 1 = Restart Auto-Negotiation Process

Table 160: PHY Control Register (Continued)
Offset: 0x00 (Hex), or 0 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
8	Duplex	R/W	See Descr	Update	Duplex Mode Selection This bit determines full-duplex or half-duplex mode if Auto-Negotiation is disabled. If Auto-Negotiation is enabled, the speed and duplex ability advertisement are located in Register 4, and Register bit 0.8 have a "don't care" definition. A write to these registers has no effect unless any one of the following also occurs: Software reset is asserted (bit 15), Power down (bit 11), or transitions from power down to normal operation. 0 = Half-duplex 1 = Full-duplex After hardware reset this bit is set as follows: If copper mode, this bit is 0.If 100BASE-FX mode, this bit is 1.
7	ColTest	RO	Always 0	Always 0	Collision Test Mode Will always be 0. The Collision test is not available, since full MII is not implemented. 0 = Disable COL signal test
6	SpeedMSB	RO	Always 0	Always 0	Speed Selection Mode (MSB) Will always be 0. 0 = 100 Mbps or 10 Mbps
5	ForceLink	R/W	0x0	0x0	Force Link Good When link is forced to be good, the link state machine is bypassed and the link is always up. 0 = Normal operation 1 = Force link good
4:0	Reserved	RO	Always 0	Always 0	Will always be 0.

Table 161: PHY Status Register
Offset: 0x01 (Hex), or 1 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	100T4	RO	Always 0	Always 0	100BASE-T4 This protocol is not available. 0 = PHY not able to perform 100BASE-T4
14	100FDX	RO	Always 1	Always 1	100BASE-T and 100BASE-X full-duplex 1 = PHY able to perform full-duplex
13	100HDX	RO	Always 1	Always 1	100BASE-T and 100BASE-X half-duplex 1 = PHY able to perform half-duplex
12	10FDX	RO	Always 1	Always 1	10BASE-T full-duplex 1 = PHY able to perform full-duplex
11	10HPX	RO	Always 1	Always 1	10BASE-T half-duplex 1 = PHY able to perform half-duplex
10	100T2FDX	RO	Always 0	Always 0	100BASE-T2 full-duplex. This protocol is not available. 0 = PHY not able to perform full-duplex
9	100T2HDX	RO	Always 0	Always 0	100BASE-T2 half-duplex This protocol is not available. 0 = PHY not able to perform half-duplex
8	ExtdStatus	RO	Always 0	Always 0	Extended Status 0 = No extended status information in Register 15
7	Reserved	RO	Always 0	Always 0	Must always be 0.
6	MFPreSup	RO	Always 1	Always 1	MF Preamble Suppression Mode Must be always 1. 1 = PHY accepts management frames with preamble suppressed
5	AnegDone	RO	0x0	0	Auto-Negotiation Complete 0 = Auto-Negotiation process not completed 1 = Auto-Negotiation process completed
4	RemoteFault	RO, LH	0x0	0	Remote Fault Mode 0 = Remote fault condition not detected 1 = Remote fault condition detected
3	AnegAble	RO	Always 1	Always 1	Auto-Negotiation Ability Mode 1 = PHY able to perform Auto-Negotiation

Table 161: PHY Status Register (Continued)
Offset: 0x01 (Hex), or 1 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
2	Link	RO, LL	0x0	0	Link Status Mode This register indicates when link was lost since the last read. For the current link status, either read this register back-to-back or read RTLink in Table 171 on page 365. 0 = Link is down 1 = Link is up
1	JabberDet	RO, LH	0x0	0	Jabber Detect 0 = Jabber condition not detected 1 = Jabber condition detected
0	ExtdReg	RO	Always 1	Always 1	Extended capability mode. 1 = Extended register capabilities

Table 162: PHY Identifier
Offset: 0x02 (Hex), or 2 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Organizationally Unique Identifier Bits 3:18	RO	0x0141	0x0141	Marvell® OUI is 0x005043 0000 0000 0101 0000 0100 0011 ^ ^ bit 1.....bit 24 Register 2.[15:0] show bits 3 to 18 of the OUI. 0000000101000001 ^ ^ bit 3.....bit 18

Table 163: PHY Identifier
Offset: 0x03 (Hex), or 3 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:10	OUI LSb	RO	Always 000011	Always 000011	Organizationally Unique Identifier bits 19:24 00 0011 ^.....^ bit 19...bit 24
9:4	ModelNum	RO	Always 001000	Always 001000	Model Number = 001000
3:0	RevNum	RO	Varies	Varies	Revision Number Contact Marvell® FAEs for information on the device revision number.

Table 164: Auto-Negotiation Advertisement Register
Offset: 0x04 (Hex), or 4 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	AnegAd NxtPage	R/W	0x0	Retain	Next Page 0 = Not advertised 1 = Advertise Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
14	Ack	RO	Always 0	Always 0	Must be 0.
13	AnegAd ReFault	R/W	0x0	Retain	Remote Fault Mode 0 = Do not set Remote Fault bit 1 = Set Remote Fault bit Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
12:11	Reserved	R/W	0x0	Retain	Must be 00. Reserved bits are R/W to allow for forward compatibility with future IEEE standards. Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
10	AnegAd Pause	R/W	0x0	Retain	Pause Mode 0 = MAC PAUSE not implemented 1 = MAC PAUSE implemented Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
9	AnegAd 100T4	R/W	0x0	Retain	100BASE-T4 mode 0 = Not capable of 100BASE-T4 Must be 0.
8	AnegAd 100FDX	R/W	0x1	Retain	100BASE-TX full-duplex Mode 0 = Not advertised 1 = Advertise Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.

Table 164: Auto-Negotiation Advertisement Register (Continued)
Offset: 0x04 (Hex), or 4 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
7	AnegAd 100HDX	R/W	0X1	Retain	100BASE-TX half-duplex Mode 0 = Not advertised 1 = Advertise Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
6	AnegAd 10FDX	R/W	0X1	Retain	10BASE-TX full-duplex Mode 0 = Not advertised 1 = Advertise Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
5	AnegAd 10HDX	R/W	0X1	Retain	10BASE-TX half-duplex Mode 0 = Not advertised 1 = Advertise Values programmed into the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (RestartAneg - Table 160 on page 350) or link goes down.
4:0	AnegAd Selector	R/W	Always 0x01	Always 0x01	Selector Field Mode 00001 = 802.3

Table 165: Link Partner Ability Register (Base Page)
Offset: 0x05 (Hex), or 5 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	LPNxt Page	RO	0x0	0	Next Page Mode Base page will be overwritten if next page is received and if Reg8NxtPg (Table 170 on page 362) is disabled. When Reg8NxtPg (Table 170 on page 362) is enabled, then next page is stored in the Link Partner Next Page register (Table 169 on page 361), and the Link Partner Ability Register (Table 165 on page 358) holds the base page. Received Code Word Bit 15 0 = Link partner not capable of next page 1 = Link partner capable of next page
14	LPAck	RO	0x0	0	Acknowledge Received Code Word Bit 14 0 = Link partner did not receive code word 1 = Link partner received link code word
13	LPRemote Fault	RO	0x0	0	Remote Fault Received Code Word Bit 13 0 = Link partner has not detected remote fault 1 = Link partner detected remote fault
12:5	LPTechAble	RO	0x00	0x00	Technology Ability Field Received Code Word Bit 12:5
4:0	LPSelector	RO	00000	00000	Selector Field Received Code Word Bit 4:0

Table 166: Link Partner Ability Register (Next Page)
Offset: 0x05 (Hex), or 5 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	LPNxtPage	RO	--	--	Next Page Mode Base page will be overwritten if next page is received and if Reg8NxtPg (Table 170 on page 362) is disabled. When Reg8NxtPg (Table 170 on page 362) is enabled, then next page is stored in the Link Partner Next Page register (Table 169 on page 361), and Link Partner Ability Register (Table 165 on page 358) holds the base page. Received Code Word Bit 15
14	LPAck	RO	--	--	Acknowledge Received Code Word Bit 14

Table 166: Link Partner Ability Register (Next Page) (Continued)
Offset: 0x05 (Hex), or 5 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
13	LPMessage	RO	--	--	Message Page Received Code Word Bit 13
12	LPack2	RO	--	--	Acknowledge 2 Received Code Word Bit 12
11	LPToggle	RO	--	--	Toggle Received Code Word Bit 11
10:0	LPData	RO	--	--	Message/Unformatted Field Received Code Word Bit 10:0

Table 167: Auto-Negotiation Expansion Register
Offset: 0x06 (Hex), or 6 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:5	Reserved	RO	Always 0x000	Always 0x000	Reserved. Must be 00000000000. The Auto-Negotiation Expansion Register is not valid until the AnegDone (Table 161 on page 353) indicates completed.
4	ParFaultDet	ROC/ LH	0x0	0x0	Parallel Detection Level 0 = A fault has not been detected via the Parallel Detection function 1 = A fault has been detected via the Parallel Detection function
3	LPNxtPg Able	RO	0x0	0x0	Link Partner Next Page Able 0 = Link Partner is not Next Page able 1 = Link Partner is Next Page able
2	LocalNxtPg Able	RO	Always 0x1	Always 0x1	Local Next Page Able This bit is equivalent to AnegAble (Table 161 on page 353). 1 = Local Device is Next Page able
1	RxNewPage	RO/LH	0x0	0x0	Page Received 0 = A New Page has not been received 1 = A New Page has been received
0	LPAnegAble	RO	0x0	0x0	Link Partner Auto-Negotiation Able 0 = Link Partner is not Auto-Negotiation able 1 = Link Partner is Auto-Negotiation able

Table 168: Next Page Transmit Register
Offset: 0x07 (Hex), or 7 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	TxNxtPage	R/W	0x0	0x0	A write to the Next Page Transmit Register implicitly sets a variable in the Auto-Negotiation state machine indicating that the next page has been loaded. Transmit Code Word Bit 15
14	Reserved	RO	0x0	0x0	Reserved Transmit Code Word Bit 14
13	TxMessage	R/W	0x1	0x1	Message Page Mode Transmit Code Word Bit 13
12	TxAck2	R/W	0x0	0x0	Acknowledge2 Transmit Code Word Bit 12
11	TxToggle	RO	0x0	0x0	Toggle Transmit Code Word Bit 11
10:0	TxData	R/W	0x001	0x001	Message/Unformatted Field Transmit Code Word Bit 10:0

Table 169: Link Partner Next Page Register
Offset: 0x08 (Hex), or 8 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	RxNxtPage	RO	0x0	0x0	If Reg8NxtPg (Table 170 on page 362) is enabled, then next page is stored in the Link Partner Next Page register (Table 169 on page 361); otherwise, theLink Partner Next Page register (Table 169 on page 361) is cleared to all 0's. Received Code Word Bit 15
14	RxAck	RO	0x0	0x0	Acknowledge Received Code Word Bit 14 0 = Link partner not capable of next page 1 = Link partner capable of next page
13	RxMessage	RO	0x0	0x0	Message Page Received Code Word Bit 13
12	RxAck2	RO	0x0	0x0	Acknowledge 2 Received Code Word Bit 12
11	RxToggle	RO	0x0	0x0	Toggle Received Code Word Bit 11
10:0	RxData	RO	0x000	0x000	Message/Unformatted Field Received Code Word Bit 10:0

Note

Registers 0x09 through 0x0F (hexadecimal (9 through 15 decimal) are reserved. Do not read or write to these registers.

Table 170: PHY Specific Control Register
Offset: 0x10 (Hex), or 16 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Enable DTE Detect	R/W	0x0	Retain	Enable DTE Detect 0 = Disable DTE Detect 1 = Enable DTE Detect
14	EDet	R/W	See Descr.	Retain	Energy Detect 0 = Disable 1 = Enable with sense and pulse Enable with sense only is not supported Enable Energy Detect takes on the appropriate value defined by the CONFIG_B pin at hardware reset. See Table 4 for details.
13	DisNLP Check	R/W	0x0	0x0	Disable Normal Linkpulse Check Linkpulse check and generation disable have no effect, if Auto-Negotiation is enabled locally. 0 = Enable linkpulse check 1 = Disable linkpulse check
12	Reg8NxtPg	R/W	0x0	0x0	Enable the Link Partner Next Page register (Table 166 on page 358) to store Next Page. If set to store next page in the Link Partner Next Page register (Table 166 on page 358), then 802.3u is violated to emulate 802.3ab. 0 = Store next page in the Link Partner Ability Register (Base Page) register (Table 165 on page 358). 1 = Store next page in the Link Partner Next Page register (Table 166 on page 358)
11	DisNLPGen	R/W	0x0	0x0	Disable Linkpulse Generation. Linkpulse check and generation disable have no effect, when Auto-Negotiation is enabled locally. 0 = Enable linkpulse generation 1 = Disable linkpulse generation
10	Reserved	RES	Always 0	Always 0	Always 0
9	DisScrambler	R/W	See Descr[1]	Retain	Disable Scrambler For 100BASE-TX, the scrambler is enabled by default and can be disabled by writing to this bit. For all other modes the scrambler is disabled regardless of the state of this bit. 0 = Enable scrambler 1 = Disable scrambler

Table 170: PHY Specific Control Register (Continued)
Offset: 0x10 (Hex), or 16 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
8	DisFEFI	R/W	CONFIG_A	Retain	Disable FEFI In 100BASE-FX mode, Disable FEFI takes on the appropriate value defined by the CONFIG_A pin at hardware reset. FEFI is automatically disabled regardless of the state of this bit if copper mode is selected. 0 = Enable FEFI 1 = Disable FEFI
7	ExtdDistance	R/W	0x0	0x0	Enable Extended Distance When using cable exceeding 100 meters, the 10BASE-T receive threshold must be lowered in order to detect incoming signals. 0 = Normal 10BASE-T receive threshold 1 = Lower 10BASE-T receive threshold
6	TPSelect	R/W	See Descr	Update	(Un)Shielded Twisted Pair This setting can be changed by writing to this bit followed by software reset. 0 = Unshielded Twisted Pair - default 1 = Shielded Twisted Pair (Un)Shielded Twisted Pair selection applies to the copper modes of operation only.
5:4	AutoMDI[X]	R/W	See Descr	Update	MDI/MDIX Crossover This setting can be changed by writing to these bits followed by software reset. 00 = Transmit on pins RXP/RXN, Receive on pins TXP/TXN 01 = Transmit on pins TXP/TXN, Receive on pins RXP/RXN 1x = Enable Automatic Crossover The default MDI/MDIX crossover setting is determined by CONFIG_B. If Auto-Crossover is enabled by CONFIG_B, then by default bits 5:4 = 11. MDI/MDIX crossover applies to the copper modes of operation only.
3:2	RxFIFO Depth	R/W	0x0	0x0	Receive FIFO Depth 00 = 4 Bytes 01 = 6 Bytes 10 = 8 Bytes 11 = Reserved
1	AutoPol	R/W	0x0	00	Polarity Reversal If Automatic polarity is disabled, then the polarity is forced to be normal in 10BASE-T mode. Polarity reversal has no effect in 100BASE-TX mode. 0 = Enable automatic polarity reversal 1 = Disable automatic polarity reversal

Table 170: PHY Specific Control Register (Continued)
Offset: 0x10 (Hex), or 16 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
0	DisJabber	R/W	0x0	00	Disable Jabber Jabber has no effect in full-duplex or in 100BASE-X mode. 0 = Enable jabber function 1 = Disable jabber function

Table 171: PHY Specific Status Register
Offset: 0x11 (Hex), or 17 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	DTE Detect Status	RO	0x0	0x0	DTE Status 0 = DTE not detected 1 = DTE detected
14	ResSpeed	RO	See Descr	Retain	Resolved Speed The values are updated after the completion of Auto-Negotiation and should be read if link is established. This bit is valid only if the resolved bit 11 is set to 1. 0 = 10 Mbps 1 = 100 Mbps If Auto-Negotiation is disabled the forced speed is already determined, so this bit is a "don't care" definition.
13	ResDuplex	RO	See Descr	Retain	Resolved Duplex Mode This value is updated after the completion of Auto-Negotiation and should be read if link is established. This bit is valid only after the resolved bit 11 is set to 1. 0 = Half-duplex 1 = Full-duplex If Auto-Negotiation is disabled the duplex is already determined, so this bit is a "don't care" definition.
12	RcvPage	RO, LH	0x0	0x0	Page Receive Mode 0 = Page not received 1 = Page received
11	Resolved	RO	0x0	0x0	Speed and Duplex Resolved. Speed and duplex bits (14 and 13) are valid only after the Resolved bit is set. The Resolved bit is set after the completion of Auto-Negotiation and should be read if link is established. 0 = Not resolved 1 = Resolved If Auto-Negotiation is disabled the speed and duplex is already determined, so this bit is a "don't care" definition.
10	RTLink	RO	0x0	0x0	Link (real time) 0 = Link down 1 = Link up
9:7	Reserved	RES	Always 000	Always 000	Always 000.
6	MDI/MDIX	RO	0x0	0x0	MDI/MDIX Crossover Status 1 = Transmit on pins TXP/TXN, Receive on pins RXP/RXN 0 = Transmit on pins RXP/RXN, Receive on pins TXP/TXN

Table 171: PHY Specific Status Register (Continued)
Offset: 0x11 (Hex), or 17 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
5	Reserved	RES	Always 0	Always 0	Always 0.
4	Sleep	RO	0x0	0x0	Energy Detect Status 0 = Chip is not in sleep mode (Active) 1 = Chip is in sleep mode (No wire activity)
3:2	Reserved	RES	Always 00	Always 00	Always 00.
1	RTPolarity	RO	0x0	0x0	Polarity (real time) 0 = Normal 1 = Reversed
0	RTJabber	RO	0x0	Retain	Jabber (real time) 0 = No Jabber 1 = Jabber

Table 172: PHY Interrupt Enable
Offset: 0x12 (Hex), or 18 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	DTE Detect State Changed Interrupt Enable	R/W	0x0	Retain	DTE Detect State Changed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
14	SpeedIntEn	R/W	0x0	Retain	Speed Changed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
13	DuplexIntEn	R/W	0x0	Retain	Duplex Changed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
12	RxPageIntEn	R/W	0x0	Retain	Page Received Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
11	AnegDone IntEn	R/W	0x0	Retain	Auto-Negotiation Completed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
10	LinkIntEn	R/W	0x0	Retain	Link Status Changed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
9	SymErrIntEn	R/W	0x0	Retain	Symbol Error Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
8	FlsCrsIntEn	R/W	0x0	Retain	False Carrier Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
7	FIFOErrInt	R/W	0x0	Retain	FIFO Over/Underflow Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
6	MDI[x]IntEn	R/W	0x0	0x0	MDI/MDIX Crossover Changed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
5	Reserved	RES	0x0	Retain	Must be 0.
4	EDetIntEn	R/W	0x0	Retain	Energy Detect Interrupt Enable 0 = Disable 1 = Enable
3:2	Reserved	RES	0x0	Retain	Must be 00.

Table 172: PHY Interrupt Enable (Continued)
Offset: 0x12 (Hex), or 18 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
1	PolarityIntEn	R/W	0x0	Retain	Polarity Changed Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable
0	JabberIntEn	R/W	0x0	Retain	Jabber Interrupt Enable 0 = Interrupt disable 1 = Interrupt enable

Table 173: PHY Interrupt Status
Offset: 0x13 (Hex), or 19 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	DTE Detect State Changed Interrupt	RO, LH	0x0	0x0	DTE detect state changed interrupt. 0 = DTE detect state not changed 1 = DTE detect state changed
14	SpeedInt	RO, LH	0x0	0x0	Speed Changed 0 = Speed not changed 1 = Speed changed
13	DuplexInt	RO, LH	0x0	0x0	Duplex Changed 0 = Duplex not changed 1 = Duplex changed
12	RxPageInt	RO, LH	0x0	0x0	0 = Page not received 1 = Page received
11	AnegDoneInt	RO, LH	0x0	0x0	Auto-Negotiation Completed 0 = Auto-Negotiation not completed 1 = Auto-Negotiation completed
10	LinkInt	RO, LH	0x0	0x0	Link Status Changed 0 = Link status not changed 1 = Link status changed
9	SymErrInt	RO, LH	0x0	0x0	Symbol Error 0 = No symbol error 1 = Symbol error
8	FlsCrsInt	RO, LH	0x0	0x0	False Carrier 0 = No false carrier 1 = False carrier
7	FIFOErrInt	RO, LH	0x0	0x0	FIFO Over /Underflow Error 0 = No over/underflow error 1 = Over/underflow error
6	MDIMDIXInt	RO, LH	0x0	0x0	MDI/MDIX Crossover Changed 0 = MDI/MDIX crossover not changed 1 = MDI/MDIX crossover changed
5	Reserved	RO	Always 0	Always 0	Always 0
4	EDetChg	RO, LH	0x0	0x0	Energy Detect Changed 0 = No Change 1 = Changed
3:2	Reserved	RO	Always 00	Always 00	Always 00

Table 173: PHY Interrupt Status (Continued)
Offset: 0x13 (Hex), or 19 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
1	PolarityInt	RO	0x0	0x0	Polarity Changed 0 = Polarity not changed 1 = Polarity changed
0	JabberInt	RO, LH	0x0	0x0	Jabber Mode 0 = No Jabber 1 = Jabber

Table 174: PHY Interrupt Port Summary (Global[1])
Offset: 0x14 (Hex), or 20 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:8	Reserved	RO	0x0	0x0	Must be 00000000.
7	Port7Int Active	RO	0x0	0x0	Port 7 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt
6	Port6Int Active	RO	0x0	0x0	Port 6 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt
5	Port5Int Active	RO	0x0	0x0	Port 5 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt
4	Port4Int Active	RO	0x0	0x0	Port 4 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt
3	Port3Int Active	RO	0x0	0x0	Port 3 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt
2	Port2Int Active	RO	0x0	0x0	Port 2 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt

Table 174: PHY Interrupt Port Summary (Global[1]) (Continued)
Offset: 0x14 (Hex), or 20 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
1	Port1Int Active	RO	0x0	0x0	Port 1 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt
0	Port0Int Active	RO	0x0	0x0	Port 0 Interrupt Active Bit is set high, if any enabled interrupt is active for the port. Bit is cleared only when all bits in register 19 are cleared. 0 = Port does not have active interrupt 1 = Port has active interrupt

1. A Global register is accessible for writing from any port.

Table 175: Receive Error Counter
Offset: 0x15 (Hex), or 21 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	RxErrCnt	RO	0x0000	0x0000	Receive Error Count This register counts receive errors on the media interface. When the maximum receive error count reaches 0xFFFF, the counter will roll over.

Table 176: LED Parallel Select Register (bits 11:0 are Global[1] bits)
Offset: 0x16 (Hex), or 22 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:12	DTE Detect Status Drop	R/W	0x4	Retain	DTE detect status drop. Once the devices no longer detect the link partner's DTE filter, the devices will wait a period of time before clearing the DTE detection status bit (17.15). The wait time is 5 seconds multiplied by the value of these bits. Example: 5*0x4 = 20 seconds
11:8	LED2	R/W	LED[1:0] 00 = LINK 01 = LINK 10 = LINK/RX 11= LINK/ ACT	Retain	LED2 Control. This is a global[1] setting. The parallel LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 0000 = COLX 0001 = ERROR 0010 = DUPLEX 0011 = DUPLEX/COLX 0100 = SPEED 0101 = LINK 0110 = TX 0111 = RX 1000 = ACT 1001 = LINK/RX 1010 = LINK/ACT 1011 = ACT (BLINK mode) 1100 = TX (Blink Mode) 1101 = RX (Blink Mode) 1110 = COLX (Blink Mode) 1111 = Force to 1 (inactive)
7:4	LED1	R/W	LED[1:0] 00 = RX 01 = ACT 10 = TX 11= DUPLEX /COLX	Retain	LED1 Control. This is a global[1] setting. The parallel LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 0000 = COLX 0001 = ERROR 0010 = DUPLEX 0011 = DUPLEX/COLX 0100 = SPEED 0101 = LINK 0110 = TX 0111 = RX 1000 = ACT 1001 = LINK/RX 1010 = LINK/ACT 1011 = ACT (BLINK mode) 1100 = TX (Blink Mode) 1101 = RX (Blink Mode) 1110 = COLX (Blink Mode) 1111 = Force to 1 (inactive)

Table 176: LED Parallel Select Register (Continued) (bits 11:0 are Global[1] bits) (Continued)
Offset: 0x16 (Hex), or 22 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
3:0	LED0	R/W	LED[1:0] 00 = TX 01 = SPEED 10 = SPEED 11= SPEED	Retain	LED0 Control. This is a global[1] setting. The parallel LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 0000 = COLX 0001 = ERROR 0010 = DUPLEX 0011 = DUPLEX/COLX 0100 = SPEED 0101 = LINK 0110 = TX 0111 = RX 1010 = LINK/ACT 1011 = ACT (BLINK mode) 1100 = TX (Blink Mode) 1101 = RX (Blink Mode) 1110 = COLX (Blink Mode) 1111 = Force to 1 (inactive)

1. Global register bits are used to control features and functions that are common to all ports in the device.

Table 177: LED Stream Select for Serial LEDs (Global Register) - 88E6097F Only
Offset: 0x17 (Hex), or 23 (Decimal)

Bits	Function	Mode	HW Rst	SW Rst	Description
15:14	LEDLnkActy	R/W	LED[1:0] 00 = Single 01 = Off 10 = Off 11 = Off	Retain	LED Link Activity The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single
13:12	LEDRcvLnk	R/W	LED[1:0] 00 = Off 01 = Single 10 = Off 11 = Off	Retain	LED Receive Link The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single
11:10	LEDActy	R/W	LED[1:0 00 = Off 01 = Off 10 = Single 11 = Off	Retain	LED Activity The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single
9:8	LEDRcv	R/W	LED[1:0] 00 = Off 01 = Off 10 = Off 11 = Single	Retain	LED Receive The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single
7:6	LEDTx	R/W	LED[1:0] 00 = Off 01 = Single 10 = Off 11 = Single	Retain	Transmit The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single
5:4	LEDLnk	R/W	LED[1:0] 00 = Off 01 = Off 10 = Single 11 = Single	Retain	Link The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single

Table 177: LED Stream Select for Serial LEDs (Global Register) - 88E6097F Only (Continued)
Offset: 0x17 (Hex), or 23 (Decimal)

Bits	Function	Mode	HW Rst	SW Rst	Description
3:2	LEDSpd	R/W	11	Retain	Speed The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single
1:0	LEDDx/ COLX	R/W	LED[1:0] 00 = Single 01 = Off 10 = Off 11 = Off	Retain	LED Duplex/ COLX The serial LED settings take on the appropriate value defined by the CONFIG_A pin at hardware reset. 00 = Off 01 = Reserved 10 = Dual 11 = Single

Table 178: PHY LED Control Register (bits 14:0 are Global[1] bits)
Offset: 0X18 (Hex), 0r 24 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	Always 0	Always 0	Must be 0.
14:12	PulseStretch	R/W	0x4	Retain	Pulse stretch duration. This is a global setting. Default Value = 100. 000 = No pulse stretching 001 = 21 ms to 42 ms 010 = 42 ms to 84 ms 011 = 84 ms to 170 ms 100 = 170 ms to 340 ms 101 = 340 ms to 670 ms 110 = 670 ms to 1.3s 111 = 1.3s to 2.7s
11:9	BlinkRate	R/W	0x1	Retain	Blink Rate. This is a global setting. Default Value = 001 000 = 42 ms 001 = 84 ms 010 = 170 ms 011 = 340 ms 100 = 670 ms 101 to 111 = Reserved
8:6	SrStrUpdate	R/W	0x2	Retain	Serial Stream Update. This is a global setting. 000 = 10 ms 001 = 21 ms 010 = 42 ms 011 = 84 ms 100 = 170 ms 101 = 340 ms 110 to 111 = Reserved
5:4	Duplex	R/W	LED[1:0] 00 = Off 01 = Single 10 = Single 11 = Single	Retain	00 = Off 01 = Reserved 10 = Dual 11 = Single
3:2	Error	R/W	11	Retain	00 = Off 01 = Reserved 10 = Dual 11 = Single
1:0	COLX	R/W	LED[1:0] 00 = Off 01 = Single 10 = Single 11 = Single	Retain	00 = Off 01 = Reserved 10 = Dual 11 = Single

1. Global register bits are used to control features and functions that are common to all ports in the device.

Table 179: PHY Manual LED Override
Offset: 0x19 (Hex), or 25 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	R/W	0x0	Retain	0
14	InvLED2	R/W	0x0	Retain	Invert LED2. This bit controls the active level of the LED2 pin for all LED options (those controlled below and those controlled by the LED Parallel Select Register at offset 0x16, decimal 22). 0 = Active Low LED2 1 = Active High LED2
13	InvLED1	R/W	0x0	Retain	Invert LED1. This bit controls the active level of the LED1 pin for all LED options (those controlled below and those controlled by the LED Parallel Select Register at offset 0x16, decimal 22). 0 = Active Low LED1 1 = Active High LED1
12	InvLED0	R/W	0x0	Retain	Invert LED0. This bit controls the active level of the LED0 pin for all LED options (those controlled below and those controlled by the LED Parallel Select Register at offset 0x16, decimal 22). 0 = Active Low LED1 1 = Active High LED1
11:10	SpLED2	R/W	0x0	Retain	LED2 Speed Select Register for LED Parallel Select register at offset 0x16, decimal 22. 00 = Normal 01 = Active for 10 Mbps Speed only 10 = Active for 100 Mbps Speed only 11 = Reserved
9:8	SpLED1	R/W	0x0	Retain	LED1 Speed Select Register for LED Parallel Select register at offset 0x16, decimal 22. 00 = Normal 01 = Active for 10 Mbps Speed only 10 = Active for 100 Mbps Speed only 11 = Reserved
7:6	SpLED0	R/W	0x0	Retain	LED0 Speed Select Register for LED Parallel Select register at offset 0x16, decimal 22. 00 = Normal 01 = Active for 10 Mbps Speed only 10 = Active for 100 Mbps Speed only 11 = Reserved
5:4	ForceLED2	R/W	0x0	Retain	00 = Normal 01 = Blink[1] 10 = LED Off 11 = LED On

Table 179: PHY Manual LED Override (Continued)
Offset: 0x19 (Hex), or 25 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
3:2	ForceLED1	R/W	0x0	Retain	00 = Normal 01 = Blink[1] 10 = LED Off 11 = LED On
1:0	ForceLED0	R/W	0x0	Retain	00 = Normal 01 = Blink[1] 10 = LED Off 11 = LED On

1. Energy Detect (Section 11.3) must be disabled on ports that are configured to blink an LED but don't have a link established.

Table 180: VCT™ Register for TXP/N Pins
Offset: 0x1A[1] (Hex), or 26 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	EnVCT	R/W, SC	0x0	0x0	Enable VCT The devices must be in forced 100 Mbps mode before enabling this bit. 0 = VCT completed 1 = Run VCT After running VCT once, bit 15 = 0 indicates VCT completed. The cable status is reported in the VCTTst bits in registers 26 and 27. Refer to the "Virtual Cable Tester®" feature on page 219.
14:13	VCTTst	RO	0x0	Retain	VCT Test Status These VCT test status bits are valid after completion of VCT. 00 = valid test, normal cable (no short or open in cable) 01 = valid test, short in cable (Impedance < 33 ohm) 10 = valid test, open in cable (Impedance > 333 ohm) 11 = Test fail
12:8	AmpRfln	RO	0x0	Retain	Amplitude of Reflection The amplitude of reflection is stored in these register bits. These amplitude bits range from 0x07 to 0x1F. 0x1F = Maximum positive amplitude 0x13 = Zero amplitude 0x07 = Maximum negative amplitude These bits are valid after completion of VCT (bit 15) and if the VCT test status bits (bits 14:13) have not indicated test failure.
7:0	DistRfln	RO	0x0	Retain	Distance of Reflection These bits refer to the approximate distance (± 1m) to the open/short location, measured at nominal conditions (room temperature and typical VDDs) These bits are valid after completion of VCT (bit 15) and if the VCT test status bits (bit 14:13) have not indicated test failure.

1. The results stored in this register apply to the Tx pin pair.

Table 181: VCT™ Register for RXP/N pins
Offset: 0x1B[1] (Hex), or 27 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	Always 0	Always 0	Reserved
14:13	VCTTst	RO	0	Retain	VCT Test Status The VCT test status bits are valid after completion of VCT. 00 = valid test, normal cable (no short or open in cable) 01 = valid test, short in cable (Impedance < 33 ohm) 10 = valid test, open in cable (Impedance > 333 ohm) 11 = Test fail
12:8	AmpRfln	RO	0	Retain	Amplitude of Reflection The amplitude of reflection is stored in these register bits. These amplitude bits range from 0x07 to 0x1F. 0x1F = Maximum positive amplitude 0x13 = Zero amplitude 0x07 = Maximum negative amplitude These bits are valid after completion of VCT (bit 15) and if VCT test status bits (bit 14:13) have not indicated test failure.
7:0	DistRfln	RO	0	Retain	Distance of Reflection These bits refer to the approximate distance (+/- 1m) to the open/short location, measured at nominal conditions (room temperature and typical VDDs) These bits are valid after completion of VCT (bit 15) and if VCT test status bits (bits 14:13) have not indicated test failure.

1. The results stored in this register apply to the Rx pin pair.

Figure 78: Cable Fault Distance Trend Line

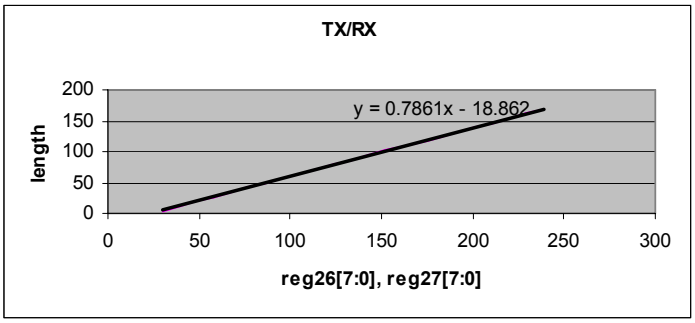


Table 182: PHY Specific Control Register II
Offset: 0x1C (Hex), or 28 (Decimal)

Bits	Field	Mode	HW Rst	SW Rst	Description
15:5	Reserved	R/W	0x0	0x0	Must be 00000000000
4	EnLineLpbk	R/W	0	Retain	0 = Disable Line Loopback 1 = Enable Line Loopback
3	SoftwareMedia-Select	R/W	0	Update	0 = Select Copper Media 1 = Select Fiber Media **NOTE:** AutoMDI and Autoneg Enable take on the values set by the hardware configuration default.
2	TDRWaitTime	R/W	0x0	Retain	0 = Wait time is 1.5s before TDR test is started 1 = Wait time is 25 ms before TDR test is started
1	EnRXCLK	R/W	0x1	Update	0 = Disable MAC interface clock (RXCLK) in sleep mode 1 = Enable MAC interface clock (RXCLK) in sleep mode
0	SelClsA	R/W	SEL_CLASS A/B[1]	Update	0 = Select Class B driver (typically used in CAT 5 applications) 1 = Select Class A driver - available for 100BASE-TX mode only (typically used in Backplane or direct connect applications, but may be used with CAT 5 applications)

1. This is the PHY SelClsA /B bit Hardware Reset name. The CONFIG_B pin of the 88E6096/88E6097/88E6097F devices contains an internal pull-up resistor, setting the default SelClsA PHY bit Hardware Reset to Class B drivers.

Note

Registers 0x1D through 0x1F (hexadecimal 29 through 31 decimal) are reserved. Do not read or write to these registers.

Section 16. Electrical Specifications

16.1 Absolute Maximum Ratings

Stresses above those listed in Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

Table 183: Absolute Maximum Ratings

Symbol	Parameter	Min	Typ	Max	Units
$V_{DD(3.3)}$	Power Supply Voltage on any 3.3V supply with respect to VSS	-0.5	3.3	+3.6	V
$V_{DD(2.5)}$	Power Supply Voltage on any 2.5V signal with respect to VSS	-0.5	2.5	+3.6 or $V_{DD(3.3)}$ +0.5[1] whichever is less	V
$V_{DD(1.5)}$	Power Supply Voltage on any 1.5V supply with respect to VSS	-0.5	1.5	+3.6 or $V_{DD(2.5)}$ +0.5[2] whichever is less	V
$V_{DD(1.2)}$	Power Supply Voltage on any 1.2V supply with respect to VSS	-0.5	1.2	+3.6 or $V_{DD(1.5)}$ +0.5[3] whichever is less	V
V_{PIN}	Voltage applied to any input pin with respect to VSS	-0.5		+3.6 or V_{DDO_PIN}[4] +0.5[5] whichever is less	V
$T_{STORAGE}$	Storage temperature	-55		+125[6]	°C

1. VDD(2.5) must never be more than 0.5V greater than VDD(3.3) or damage will result. Power must be applied to VDD(3.3) before or at the same time as VDD(2.5).
2. VDD(1.5) must never be more than 0.5V greater than VDD(2.5) or damage will result. Power must be applied to VDD(2.5) before or at the same time as VDD(1.5).
3. VDD(1.2) must never be more than 0.5V greater than VDD(1.5) or damage will result. Power must be applied to VDD(1.5) before or at the same time as VDD(1.2).
4. The VDDO pad ring has separate I/O power supply options. Therefore, the voltage applied to a group of I/O pins must follow what is defined in Section 1.
5. VPIN must never be more than 0.5V greater than VDDO or damage will result.
6. 125°C is the re-bake temperature. For extended storage time greater than 24 hours, +85°C should be the maximum.

16.2 Recommended Operating Conditions

Table 184: Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
$V_{DD(3.3)}$	3.3V power supply	For any 3.3V supply pin	3.135	3.3	3.465	V
$V_{DD(2.5)}$	2.5V power supply	For any 2.5V supply pin[1] - Commercial parts	2.375	2.5	2.730	V
		For any 2.5V supply pin - Industrial parts	2.470	2.6		
$V_{DD(1.5)}$	1.5V power supply	For any 1.5V supply pin	1.425	1.5	1.590	V
$V_{DD(1.2)}$	1.2V power supply	For any 1.2V supply pin - Commercial parts	1.140	1.2	1.365	V
		For any 1.2V supply pin - Industrial parts	1.235	1.3		
T_A	Ambient operating temperature[2]	Commercial parts	0		70	°C
		Industrial parts[3]	-40		85	°C
T_J	Maximum junction temperature				125[2]	°C
IREF	Internal bias reference	External resistor value required to be placed between IREF and VSS pins	1980	2000	2020	Ω

1. Some VDDO pins can be set to either 2.5V or 3.3V. To guarantee proper operation they must be set within the appropriate ranges in this table. VDDO voltages between 2.625V and 3.135V are not supported.
2. The important parameter is maximum junction temperature. As long as the maximum junction temperature is not exceeded, the device can be operated at any ambient temperature. Refer to White Paper on "TJ Thermal Calculations" for more information.
3. Industrial part numbers have an "I" following the commercial part numbers. See "Ordering Part Numbers and Package Markings" on page 417.

16.3 Thermal Conditions for 88E6096/88E6097 devices 176-pin TQFP Package

Symbol	Parameter	Condition	Min	Typ	Max	Units
θ_{JA}	Thermal resistance[1] - junction to ambient of the 88E6096/88E6097 device 176-Pin TQFP package $\theta_{JA} = (T_J - T_A)/ P$ P = Total Power Dissipation	JEDEC 3 in. x 4.5 in. 4-layer PCB with no air flow		20.30		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 1 meter/ sec air flow		16.80		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 2 meter/ sec air flow		15.80		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 3 meter/ sec air flow		15.20		°C/W
ψ_{JT}	Thermal characteristic parameter[1] - junction to top center of the 88E6096/ 88E6097 device 176-Pin TQFP package $\psi_{JT} = (T_J - T_{TOP})/P$. T_{TOP} = Temperature on the top center of the package	JEDEC 3 in. x 4.5 in. 4-layer PCB with no air flow		0.13		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 1 meter/ sec air flow		0.21		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 2 meter/ sec air flow		0.27		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 3 meter/ sec air flow		0.31		°C/W
θ_{JC}	Thermal resistance[1] - junction to case of the 88E6096/ 88E6097 device 176-Pin TQFP package $\theta_{JC} = (T_J - T_C)/ P_{Top}$ P_{Top} = Power Dissipation from the top of the package	JEDEC with no air flow		5.20		°C/W
θ_{JB}	Thermal resistance[1] - junction to board of the 88E6096/ 88E6097 device 176-Pin TQFP package $\theta_{JB} = (T_J - T_B)/ P_{bottom}$ P_{bottom} = power dissipation from the bottom of the package to the PCB surface.	JEDEC with no air flow		10.60		°C/W

1. Refer to white paper on TJ Thermal Calculations for more information.

16.4 Thermal Conditions for 88E6097F device 216-pin LQFP Package

Symbol	Parameter	Condition	Min	Typ	Max	Units
θ_{JA}	Thermal resistance[1] - junction to ambient of the 88E6097F device 216-Pin LQFP package $\theta_{JA} = (T_J - T_A)/ P$ P = Total Power Dissipation	JEDEC 3 in. x 4.5 in. 4-layer PCB with no air flow		19.60		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 1 meter/ sec air flow		16.00		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 2 meter/ sec air flow		15.10		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 3 meter/ sec air flow		14.50		°C/W
ψ_{JT}	Thermal characteristic parameter[1] - junction to top center of the 88E6097F device 216-Pin LQFP package $\psi_{JT} = (T_J - T_{TOP})/P$. T_{TOP} = Temperature on the top center of the package	JEDEC 3 in. x 4.5 in. 4-layer PCB with no air flow		0.33		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 1 meter/ sec air flow		0.51		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 2 meter/ sec air flow		0.62		°C/W
		JEDEC 3 in. x 4.5 in. 4-layer PCB with 3 meter/ sec air flow		0.70		°C/W
θ_{JC}	Thermal resistance[1] - junction to case of the 88E6097F device 216-Pin LQFP package $\theta_{JC} = (T_J - T_C)/ P_{Top}$ P_{Top} = Power Dissipation from the top of the package	JEDEC with no air flow		7.60		°C/W
θ_{JB}	Thermal resistance[1] - junction to board of the 88E6097F device 216-Pin LQFP package $\theta_{JB} = (T_J - T_B)/ P_{bottom}$ P_{bottom} = power dissipation from the bottom of the package to the PCB surface.	JEDEC with no air flow		10.00		°C/W

1. Refer to white paper on TJ Thermal Calculations for more information.

16.5 Current Consumption

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 185: Current Consumption

Pins	Parameter	Condition	Min	Typ	Max	Units
V_{DDO}[1]	Outputs with 3.3V power applied	All ports (FE and GE) idle		96		mA
		All ports (FE and GE) active		96		mA
	Outputs with 2.5V power applied	All ports (FE and GE) idle		41		mA
		All ports (FE and GE) active		41		mA
V_{DDAH}[2]	2.5V analog power to PHY (Energy Detect Disabled)	No link on any FE PHY		86		mA
		All FE PHYs 10 Mbps linked and idle		212		mA
		All FE PHYs 10 Mbps and active		215		mA
		All FE PHYs 100 Mbps		180		mA
V_{CT}	2.5V power to FE Magnetics Center Tap	No link on any FE PHY		2		mA
		All FE PHYs 10 Mbps linked and idle		2		mA
		All FE PHYs 10 Mbps and active		475		mA
		All FE PHYs 100 Mbps		159		mA
Px_V_{DDAH}	2.5V analog power to SERDES	All GE SERDES active		80		mA
		All GE SERDES power down		0		mA
Px_V_{TT}	1.5V analog power to SERDES	All GE SERDES active		64		mA
		All GE SERDES power down		0		mA
	1.2V analog power to SERDES	All GE SERDES active		39		mA
		All GE SERDES power down		0		mA
V_{DDAL}	1.2V analog power to PHY	No link on any FE PHY		4		mA
		All FE PHYs 10 Mbps linked and idle		4		mA
		All FE PHYs 10 Mbps and active		6		mA
		All FE PHYs 100 Mbps		50		mA
V_{DD_CORE}	1.2V analog power to core	All ports (FE and GE) idle		252		mA
		All ports (FE and GE) active		290		mA

1. VDDO includes VDDO_P9, VDDO_P10, VDDO_LED, VDDO_PHY, VDDO_EEPROM
2. VDDAH includes VDDAH and VDD_PLL

16.6 DC Electrical Characteristics

16.6.1 Digital Operating Conditions

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 186: Digital Operating Conditions

Symbol	Parameter	Pins	Condition	Min	Typ	Max	Units
V_{IH}	High level input voltage	All pins	VDDO = 3.135V	2.0			V
			VDDO = 2.375V	1.7			V
V_{IL}	Low level input voltage	All pins	VDDO = 3.135V	-0.3		0.8	V
			VDDO = 2.375V	-0.3		0.7	V
V_{OH}	High level output voltage	LED pins	I_{OH} = -8 mA	VDDO - 0.4			V
		All others (except INTn [1])	I_{OH} = -4 mA	VDDO - 0.4			V
V_{OL}	Low level output voltage	INTn and LED pins	I_{OL}= 8 mA			0.4	V
		All others	I_{OL}= 4 mA			0.4	V
I_{ILK}	Input leakage current	With pull-up resistor	$0<V_{IN}<V_{DD}$			+ 10 - 50	µA
		With pull-down resistor	$0<V_{IN}<V_{DD}$			+ 50 - 10	µA
		All others	$0<V_{IN}<V_{DD}$			±10	µA
C_{IN}	Input capacitance	All pins				5	pF

1. The INTn is an active low, open drain pin. See INTn description in the Signal Description.

16.6.2 SERDES Electrical Specifications

Table 187: Transmitter DC Characteristics with Px_VTT = 1.5V Typical

Symbol	Parameter[1]	Min	Typ	Max	Units
V_{OH}	Output Voltage High			1500	mV
V_{OL}	Output Voltage Low	800			mV
V_{RING}	Output Ringing			10	mV
$\|V_{OD}\|$	Output Voltage Swing (single - ended, peak-to-peak)	100		700	mV peak
V_{OS}	Output Offset Voltage	1175		1425	mV
R_O	Output Impedance (single-ended) (50 ohm termination)	40		60	Ωs
Delta R_O	Mismatch in a pair			10	%
Delta V_{OD}	Change in VOD between 0 and 1			25	mV
Delta V_{OS}	Change in VOS between 0 and 1			25	mV
I_{SA}, I_{SB}	Output current on short to VSS			30	mA
I_{SAB}	Output current when a, b are shorted			12	mA
I_{XA}, I_{XB}	Power off leakage current			5	mA

1. All parameters measured at RLOAD = 100 Ω ± 1% load (P[9:0]_TXP/N).

Table 188: Transmitter DC Characteristics with Px_VTT = 1.2V Typical

Symbol	Parameter[1]	Min	Typ	Max	Units
V_{OH}	Output Voltage High			1200	mV
V_{OL}	Output Voltage Low	800			mV
V_{RING}	Output Ringing			10	mV
$\|V_{OD}\|$	Output Voltage Swing (single - ended, peak-to-peak)	100		400	mV peak
V_{OS}	Output Offset Voltage	950		1150	mV
R_O	Output Impedance (single-ended) (50 ohm termination)	40		60	Ωs
Delta R_O	Mismatch in a pair			10	%
Delta V_{OD}	Change in VOD between 0 and 1			25	mV
Delta V_{OS}	Change in VOS between 0 and 1			25	mV
I_{SA}, I_{SB}	Output current on short to VSS			24	mA
I_{SAB}	Output current when a, b are shorted			12	mA
I_{XA}, I_{XB}	Power off leakage current			5	mA

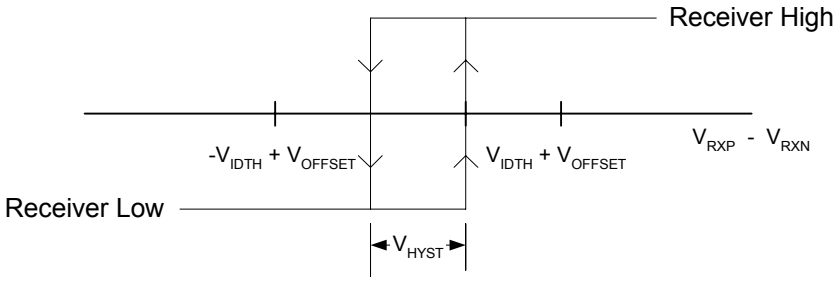
1. All parameters measured at RLOAD = 100 Ω ± 1% load (P[9:0]_TXP/N).

Table 189: Receiver DC Characteristics

Symbol	Parameter	Min	Typ	Max	Units
V_I	Input Voltage range a or b	675		1725	mV
V_{IDTH}[1]	Input Differential Threshold |RXP - RXN|	50		200	mV
V_{HYST}[1]	Input Differential Hysteresis	25			mV
R_{IN}	Receiver 100Ω Differential Input Impedance	80	100	120	Ω

1. Receiver is at high level when VRXP - VRXN is greater than VIDTH (min) + VOFFSET and is at low level when VRXP - VRXN is less than -VIDTH (min) + VOFFSET. A minimum hysteresis of VHYST is present between -VIDTH and +VIDTH as shown in Figure 79. When the fiber link is down, an offset of VOFFSET is applied to prevent false signal detect due to noise. When the fiber link is up, the offset circuit is disabled.

Figure 79: Input Differential Hysteresis



16.6.3 IEEE DC Transceiver Parameters

IEEE tests are typically based on templates and cannot simply be specified by a number. For an exact description of the template and the test conditions, refer to the IEEE specifications:
-10BASE-T IEEE 802.3 Clause 14
-100BASE-TX ANSI X3.263-1995
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 190: IEEE DC Transceiver Parameters

Symbol	Parameter	Pins	Condition	Min	Typ	Max	Units
V_{ODIFF}	Absolute peak differential output voltage	TXP/N[7:0]	10BASE-T no cable	2.2	2.5	2.8	V
		TXP/N[7:0]	10BASE-T cable model	585[1]			mV
		TXP/N[1:0]	100BASE-FX mode	0.4	0.8	1.2	V
		TXP/N[7:0]	100BASE-TX mode	0.950	1.0	1.05	V
	Overshoot[2]	TXP/N[7:0]	100BASE-TX mode	0		5%	V
	Amplitude Symmetry (positive/ negative)	TXP/N[7:0]	100BASE-TX mode	0.98x		1.02x	V+/V-
V_{IDIFF}	Peak Differential Input Voltage accept level	RXP/N[7:0]	10BASE-T mode	585[3]			mV
		RXP/N[1:0] P[1:0]_SDET	100BASE-FX mode	200			mV
	Signal Detect Assertion	RXP/N[1:0]	100BASE-TX mode	1000	460[4]		mV peak-peak
	Signal Detect De-assertion	RXP/N[1:0]	100BASE-TX mode	200	360[5]		mV peak-peak

1. IEEE 802.3 Clause 14, Figure 14.9 shows the template for the “far end” wave form. This template allows as little as 495 mV peak differential voltage at the far end receiver.
2. ANSI X3.263-1995 Figure 9-1.
3. The input test is actually a template test. IEEE 802.3 Clause 14, Figure 14.17 shows the template for the receive wave form.
4. The ANSI TP-PMD specification requires that any received signal with peak-to-peak differential amplitude greater than 1000 mV should turn on signal detect (internal signal in 100BASE-TX mode). The devices will accept signals typically with 460 mV peak-to-peak differential amplitude.
5. The ANSI TP-PMD specification requires that any received signal with peak-to-peak differential amplitude less than 200 mV should be de-assert signal detect (internal signal in 100BASE-TX mode). The devices will reject signals typically with peak-to-peak differential amplitude less than 360 mV.

16.7 AC Electrical Specifications

16.7.1 Receiver AC Characteristics

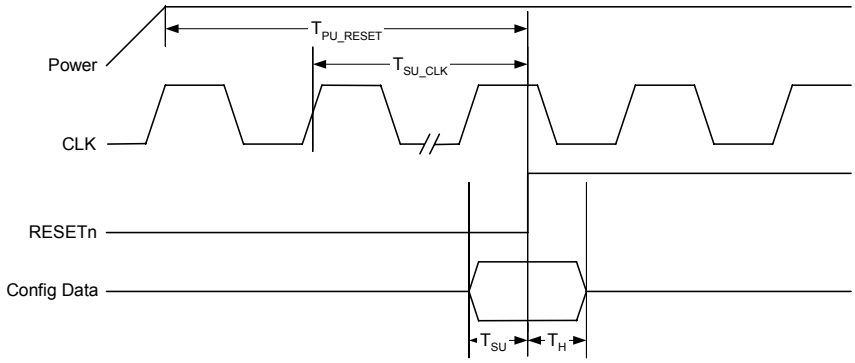
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified.)

Table 191: Reset and Configuration Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_{PU_RESET}	Valid power to RESETn de-asserted or RESETn assertion time	At power up or sub-sequent resets after power up	10			ms
T_{SU_CLK}	Number of valid REFCLK cycles prior to RESETn de-asserted		10			Clks
T_{SU}	Configuration data valid prior to RESETn de-asserted[1]		200			ns
T_{HD}	Config data valid after RESETn de-asserted		0			ns

1. When RESETn is low all configuration pins become inputs, and the value seen on these pins is latched on the rising edge of RESETn. All configuration pins that become outputs during normal operation will remain tri-stated for 40 ns after the rising edge of RESETn.

Figure 80: Reset and Configuration Timing



16.7.2 Clock Timing

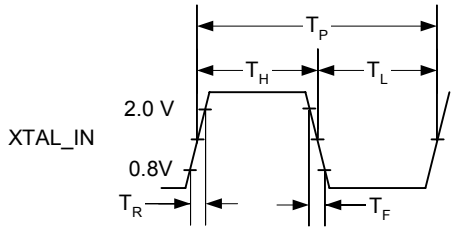
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified.)

Table 192: Clock Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_P[1]	XTAL_IN period		40 -50 ppm	40	40 +50 ppm	ns
T_H	XTAL_IN high time		16			ns
T_L	XTAL_IN low time		16			ns
T_R	XTAL_IN rise				3	ns
T_F	XTAL_IN fall				3	ns

1. 25.000 MHz

Figure 81: Oscillator Clock Timing



16.8 GMII Timing

16.8.1 GMII Transmit Timing

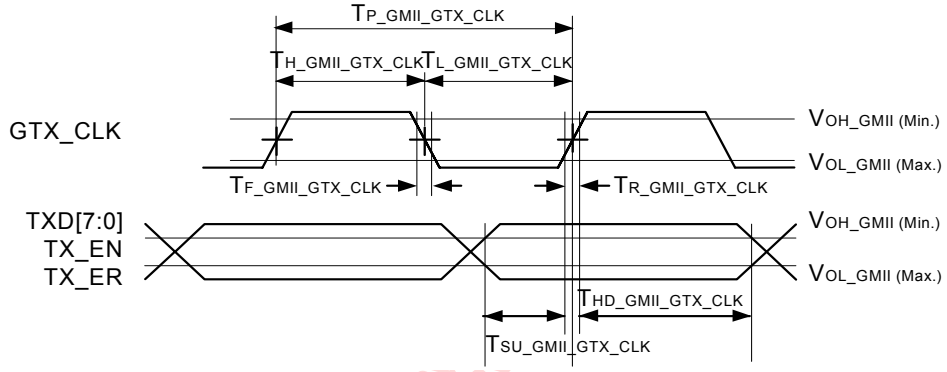
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 193: GMII Transmit Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
$T_{SU\_GMII\_GTX\_CLK}$	GMII output to clock		2.5			ns
$T_{HD\_GMII\_GTX\_CLK}$	GMII clock to output		0.5			ns
$T_{H\_GMII\_GTX\_CLK}$	GTX_CLK High		2.5[1]		5.5	ns
$T_{L\_GMII\_GTX\_CLK}$	GTX_CLK Low		2.5[1]		5.5	ns
$T_{P\_GMII\_GTX\_CLK}$	GTX_CLK Period		7.5[1]	8.0		ns
$T_{R\_GMII\_GTX\_CLK}$	GTX_CLK Rise Time				1.0	ns
$T_{F\_GMII\_GTX\_CLK}$	GTX_CLK Fall Time				1.0	ns
$T_{RSLEW\_GMII\_GTX\_CLK}$	GTX_CLK Rising Slew Rate		0.6[2]			V/ns
$T_{FSLEW\_GMII\_GTX\_CLK}$	GTX_CLK Falling Slew Rate		0.6[2]			V/ns

1. GTX_CLK numbers not guaranteed during transition between 10/100/1000BASE-T operation.
2. Instantaneous change during internal VIH_GMII (Min.) and VIL_GMII (Max.).

Figure 82: GMII Transmit Timing



16.8.2 GMII Receive Timing

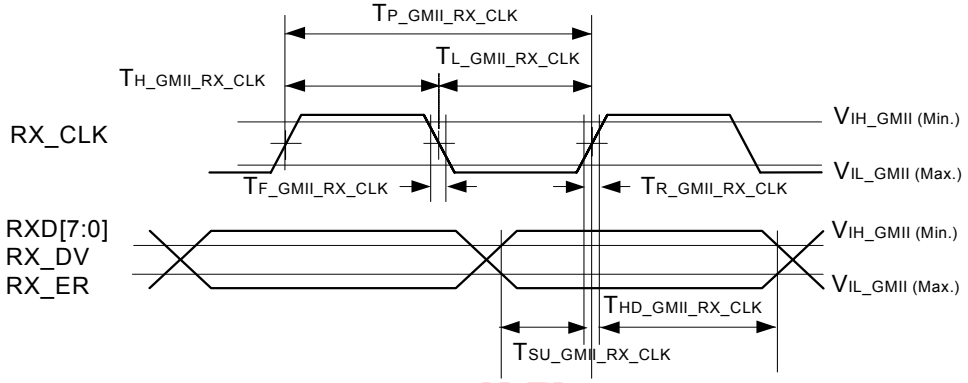
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 194: GMII Receive Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
$T_{SU\_GMII\_RX\_CLK}$	GMII Setup Time		2.0			ns
$T_{HD\_GMII\_RX\_CLK}$	GMII Hold Time		0			ns
$T_{H\_GMII\_RX\_CLK}$	RX_CLK High		2.5[1]			ns
$T_{L\_GMII\_RX\_CLK}$	RX_CLK Low		2.5[1]			ns
$T_{P\_GMII\_RX\_CLK}$	RX_CLK Period		7.5[1]	8.0	8.5	ns
$F_{GMII\_RX\_CLK}$	RX_CLK Frequency		125[1] -100 ppm		125 +100 ppm	MHz
$T_{R\_GMII\_RX\_CLK}$	RX_CLK Rise Time				1.0	ns
$T_{F\_GMII\_RX\_CLK}$	RX_CLK Fall Time				1.0	ns

1. RX_CLK toggle rate is “don’t care” if link is down, or if not in 1000BASE-T mode.

Figure 83: GMII Receive Timing



16.9 RGMII Timing

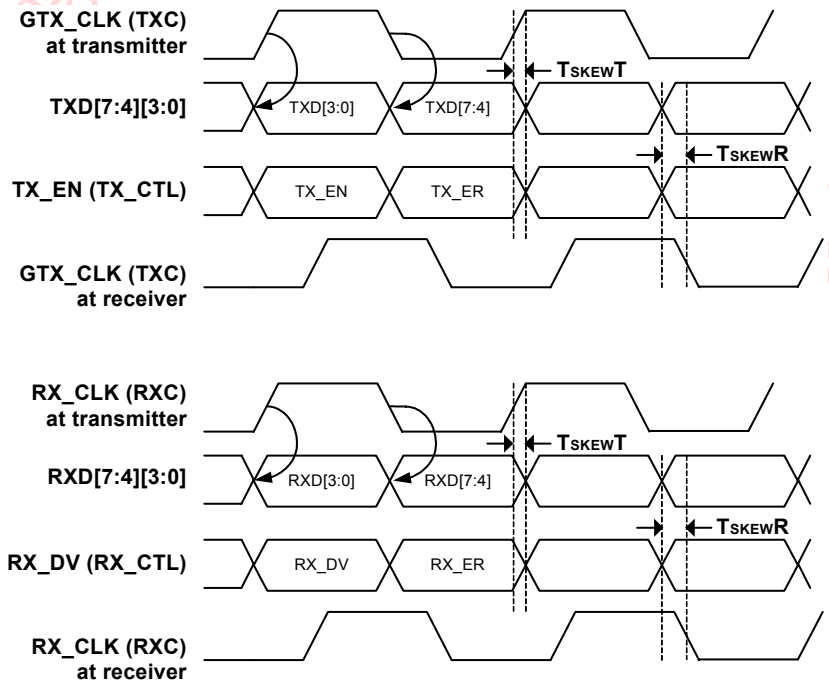
Table 195: RGMII Interface Timing

(For other timing modes see Section 16.9.1 "RGMII Timing for Different RGMII Modes" on page 397.)

Symbol	Parameter	Min	Typ	Max	Units
TskewT	Data to Clock output Skew (at transmitter)	-500	0	500	ps
TskewR	Data to Clock input Skew (at receiver)	1.0	-	2.6	ns
T_{CYCLE}	Clock Cycle Duration	7.2	8.0	8.8	ns
$T_{CYCLE\_HIGH1000}$	High Time for 1000BASE-T [1]	3.6	4.0	4.4	ns
T_{RISE}/T_{FALL}	Rise/Fall Time (20-80%)			0.75	ns

1. Duty cycle may be stretched/shrunk during speed changes or while transitioning to a received packet's clock domain as long as minimum duty cycle is not violated and stretching occurs for no more than three TCYCLE of the lowest speed transitioned between.

Figure 84: RGMII Multiplexing and Timing



16.9.1 RGMII Timing for Different RGMII Modes

16.9.1.1 RGMII Transmit Timing

Table 196: Transmit - TXC Timing when Device Addr 0x16, Offset 0x1A bit [9] = 0
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Symbol	Parameter	Min	Typ	Max	Units
t_{sskew}	Device Addr 0x16, Offset 0x1A bit [9] = 0	-0.5		0.5	ns

Figure 85: Transmit - TXC Timing when Device Addr 0x16, Offset 0x1A bit [9] = 0

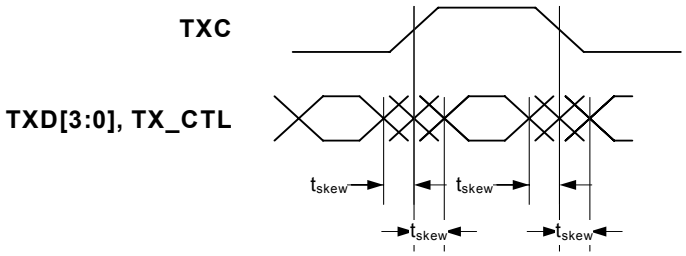
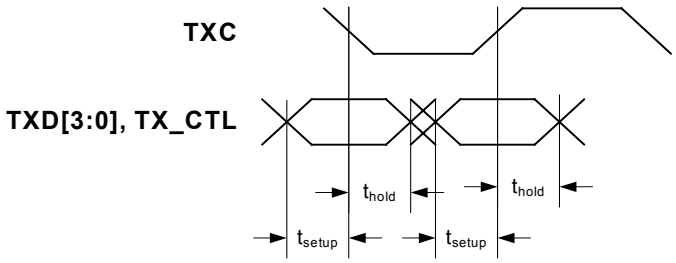


Table 197: Transmit - TXC Timing when Device Addr 0x16, Offset 0x1A bit [9] = 1
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Symbol	Parameter	Min	Typ	Max	Units
t_{setup}	Device Addr 0x16, Offset 0x1A bit [9] = 1	1.2			ns
t_{hold}		1.0			ns

Figure 86: Transmit - TXC Timing when Device Addr 0x16, Offset 0x1A bit [9] = 1



16.9.1.2 RGMII Receive Timing

Table 198: Receive - RXC Timing when Device Addr 0x16, Offset 0x1A bit [10] = 0
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Symbol	Parameter	Min	Typ	Max	Units
t_{setup}	Device Addr 0x16, Offset 0x1A bit [10] = 0	1.0			ns
t_{hold}		0.8			ns

Figure 87: Receive - RXC Timing when Device Addr 0x16, Offset 0x1A bit [10] = 0

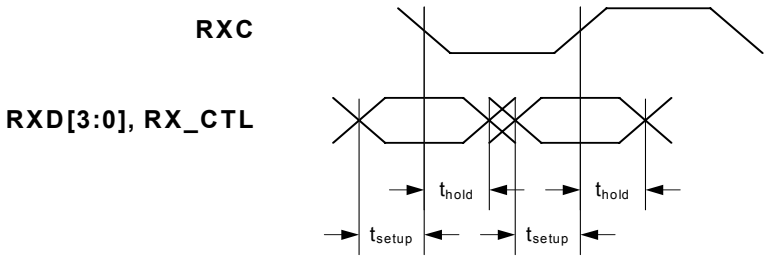
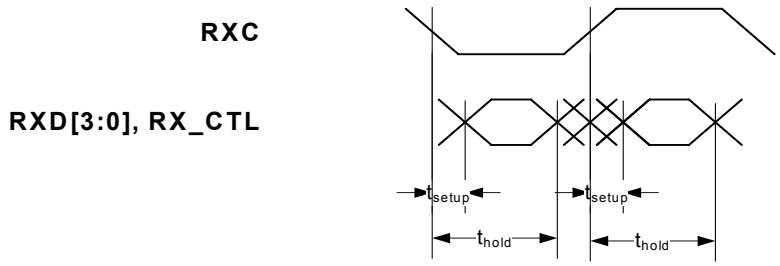


Table 199: Receive - RXC Timing when Device Addr 0x16, Offset 0x1A bit [10] = 1
(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Symbol	Parameter	Min	Typ	Max	Units
t_{setup}	Device Addr 0x16, Offset 0x1A bit [10] = 1	-0.9			ns
t_{hold}		2.7			ns

Figure 88: Receive - RXC Timing when Device Addr 0x16, Offset 0x1A bit [10] = 1



16.10 MII Timing

16.10.1 MII MAC Mode Clock Timing

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

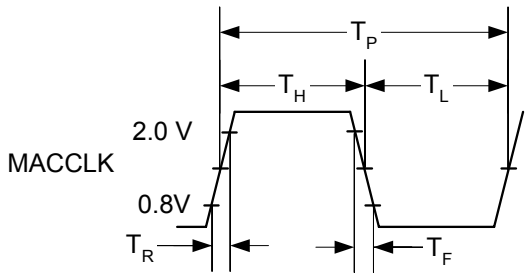
In MII MAC mode, the P9_RXCLK, P10_RXCLK and P9_TXCLK, P10_TXCLK pins are inputs.

Table 200: MII MAC Mode Clock Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_P	MACCLK_IN period		20[1]	40 or 400		ns
T_H	MACCLK_IN high time		8			ns
T_L	MACCLK_IN low time		8			ns
T_R	MACCLK_IN rise				3	ns
T_F	MACCLK_IN fall				3	ns

1. This value applies for 200 Mbps mode

Figure 89: MAC Clock Timing



16.10.2 MII Receive Timing - MAC Mode

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

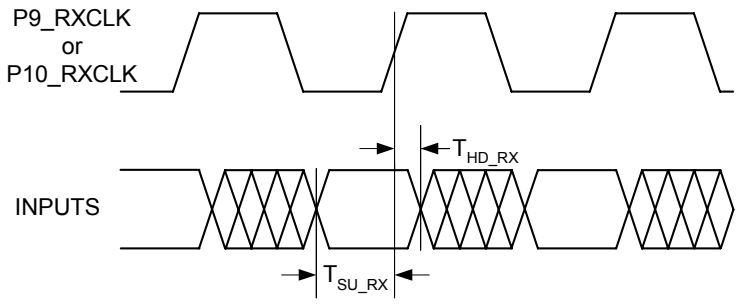
Table 201: MII Receive Timing—MAC Mode 100 Mbps Operation

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_{SU_RX}	MII inputs (P9_RXD[3:0], P9_RXDV, and P9_RXER) valid prior to P9_RXCLK going high, or MII inputs (P10_RXD[3:0], P10_RXDV, and P10_RXER) valid prior to P10_RXCLK going high	With 10 pF load	10			ns
T_{HD_RX}	MII inputs (P9_RXD[3:0], P9_RXDV, and P9_RXER) valid after P9_RXCLK going high, or MII inputs (P10_RXD[3:0], P10_RXDV, and P10_RXER) valid after P10_RXCLK going high	With 10pF load	10			ns

Table 202: MII Receive Timing—MAC Mode 200 Mbps Operation

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_{SU_RX}	MII inputs (P9_RXD[3:0], P9_RXDV, and P9_RXER) valid prior to P9_RXCLK going high, or MII inputs (P10_RXD[3:0], P10_RXDV, and P10_RXER) valid prior to P10_RXCLK going high	With 10 pF load	5			ns
T_{HD_RX}	MII inputs (P9_RXD[3:0], P9_RXDV, and P9_RXER) valid after P9_RXCLK going high, or MII inputs (P10_RXD[3:0], P10_RXDV, and P10_RXER) valid after P10_RXCLK going high	With 10pF load	2			ns

Figure 90: MAC Mode MII Receive Timing



16.10.3 MII Transmit Timing - MAC Mode

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 203: MII Transmit Timing—MAC Mode 100 Mbps Operation

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_{TXCLK}	P9_TXCLK period P10_TXCLK period	10BASE mode[1]		400		ns
		100BASE mode		40		ns
T_{H_TXCLK}	P9_TXCLK high P10_TXCLK high	10BASE mode		200		ns
		100BASE mode		20		ns
T_{L_TXCLK}	P9_TXCLK low P10_TXCLK low	10BASE mode		200		ns
		100BASE mode		20		ns
T_{CQ_MAX}	P9_TXCLK to outputs (P9_TXD[3:0], P9_TXEN) valid P10_TXCLK to outputs (P10_TXD[3:0], P10_TXEN) valid	With 10 pF load			25	ns
T_{CQ_MIN}	P9_TXCLK to outputs P9_TXD[3:0], P9_TXEN) invalid P10_TXCLK to outputs P10_TXD[3:0], P10_TXEN) invalid	With 10 pF load	0			ns

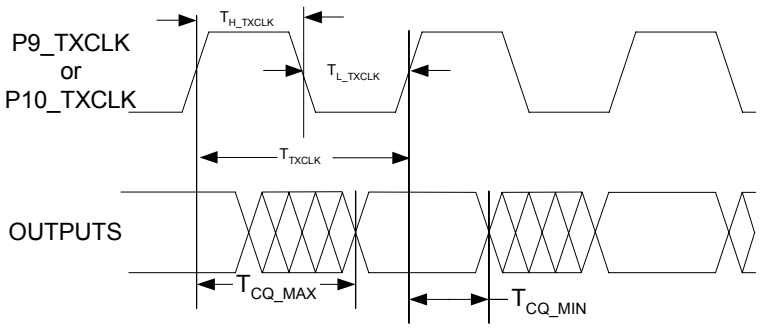
1. 2.5 MHz for 10 Mbps or 25 MHz for 100 Mbps

Table 204: MII Transmit Timing—MAC Mode 200 Mbps Operation

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_{TXCLK}	P9_TXCLK period P10_TXCLK period	200 Mbps Operation mode[1]		20		ns
T_{H_TXCLK}	P9_TXCLK high P10_TXCLK high	200 Mbps Operation		10		ns
T_{L_TXCLK}	P9_TXCLK low P10_TXCLK low	200 Mbps Operation		10		ns
T_{CQ_MAX}	P9_TXCLK to outputs (P9_TXD[3:0], P9_TXEN) valid P10_TXCLK to outputs (P10_TXD[3:0], P10_TXEN) valid	With 10 pF load			15	ns
T_{CQ_MIN}	P9_TXCLK to outputs P9_TXD[3:0], P9_TXEN) invalid P10_TXCLK to outputs P10_TXD[3:0], P10_TXEN) invalid	With 10 pF load	3			ns

1. 2.5 MHz for 10 Mbps or 25 MHz for 100 Mbps

Figure 91: MAC Mode MII Transmit Timing



16.11 Serial Management Interface (SMI) Timing

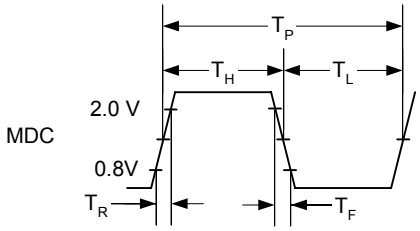
16.11.1 SMI Clock Timing (CPU Set)

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 205: SMI Clock Timing (CPU Set)

Symbol	Parameter	Condition	Min	Typ	Max	Units	Notes
T_P	MDC period		120			ns	8.33 MHz
T_H	MDC high time		48			ns	
T_L	MDC low time		48			ns	
T_R	MDC rise				6	ns	
T_F	MDC fall				6	ns	

Figure 92: SMI Clock Timing (CPU Set)



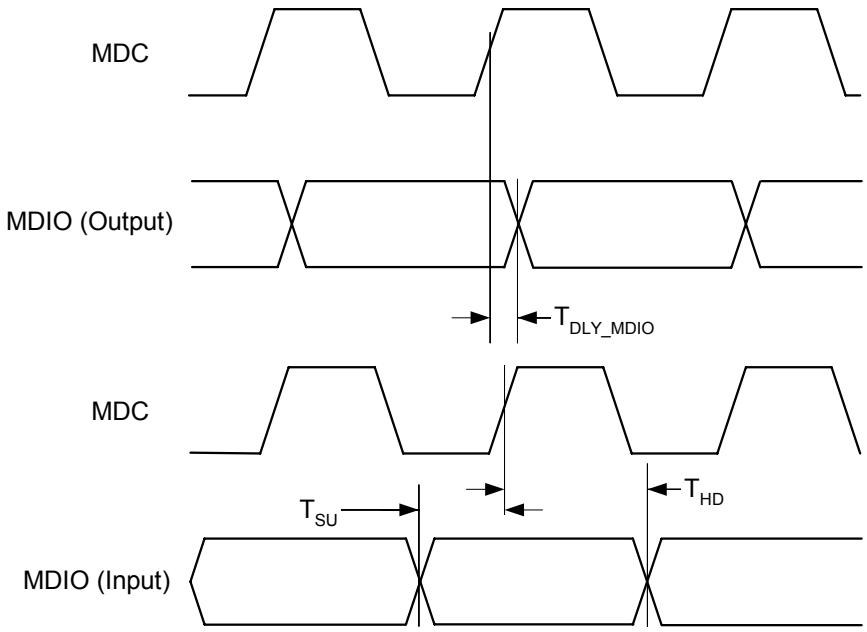
16.11.2 SMI Data Timing (CPU Set)

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 206: SMI Data Timing (CPU Set)

Symbol	Parameter	Condition	Min	Typ	Max	Units	Notes
T_{DLY_MDIO}	MDC to MDIO (Output) delay time		0		12.5	ns	
T_{SU}	MDIO (Input) to MDC setup time		10			ns	
T_{HD}	MDIO (Input) to MDC hold time		10			ns	

Figure 93: SMI Data Timing



16.11.3 SMI Timing (PHY Set)

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 207: SMI Timing (PHY Set) - (PPU_EN = 1)

Symbol	Parameter	Condition	Min	Typ	Max	Units	Notes
T_P	MDC period		120[1]			ns	8.33 MHz
T_H	MDC high time		48			ns	
T_L	MDC low time		48			ns	
T_R	MDC rise				6	ns	
T_F	MDC fall				6	ns	
T_{RXSU}	MDIO input setup time		12.5			ns	
T_{RXHD}	MDIO input hold time		0			ns	
T_{TXSU}	MDIO output setup time		10			ns	
T_{TXHD}	MDIO output hold time		10			ns	

1. MDC_PHY will track MDC_CPU when the PPU is disabled. When the PPU is enabled the MDC_PHY period will be 240 ns.

Figure 94: SMI Timing Input (PHY Mode)

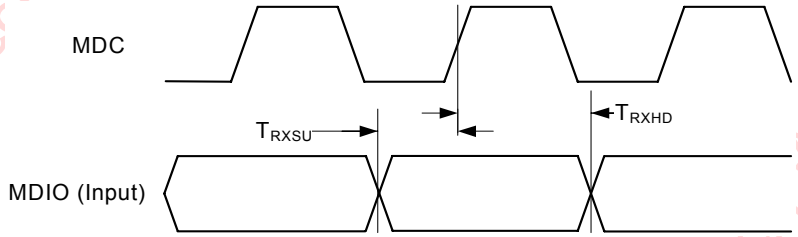
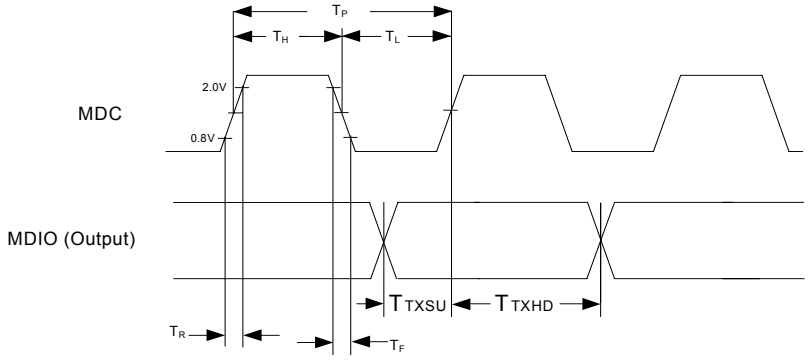


Figure 95: SMI Timing Output (PHY Mode)



(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 208: SMI Direct Access Mode - (PPU_EN = 0)

Symbol	Parameter	Condition	Min	Typ	Max	Units	Notes
T_{skew_clk}			3		8.5	ns	
$T_{skew\_cpu2phy}$			3		8.5	ns	
$T_{skew\_phy2cpu}$			3		8.5	ns	

Figure 96: SMI Direct Access Mode Configuration

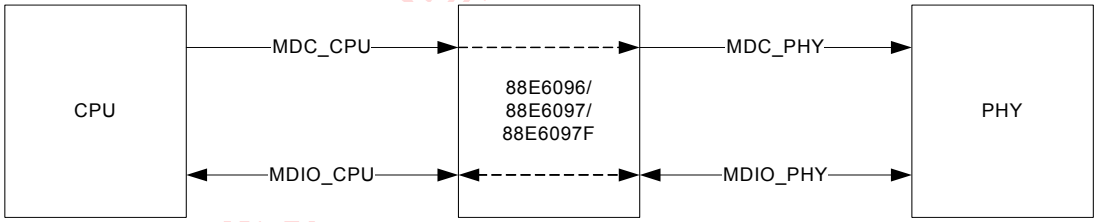


Figure 97: SMI Direct Access MDC CPU to PHY Timing Skew

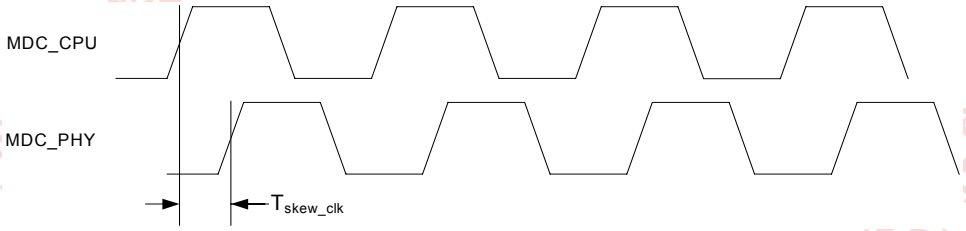


Figure 98: SMI Direct Access MDIO CPU to PHY Timing Skew

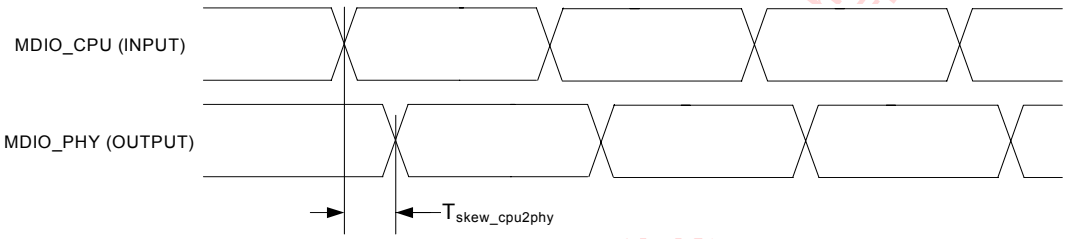
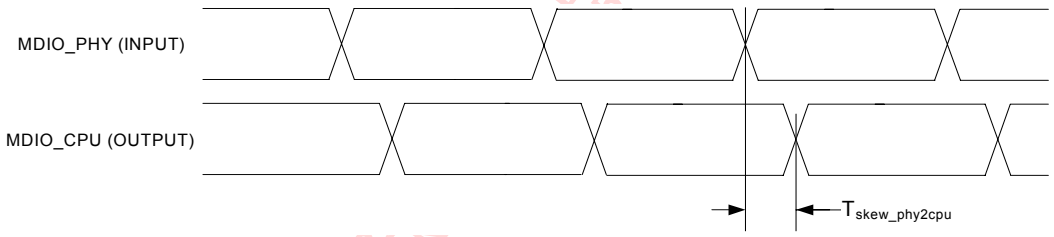


Figure 99: SMI Direct Access MDIO PHY to CPU Timing Skew



16.12 EEPROM Timing

16.12.1 2-Wire EEPROM Timing

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 209: 2-Wire EEPROM Input Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_P	EE_CLK period			19200		ns
T_H	EE_CLK high time			9600		ns
T_L	EE_CLK low time			9600		ns
T_{IN}	EE_CLK input time		50		4500	ns

Figure 100:2-Wire Input Timing

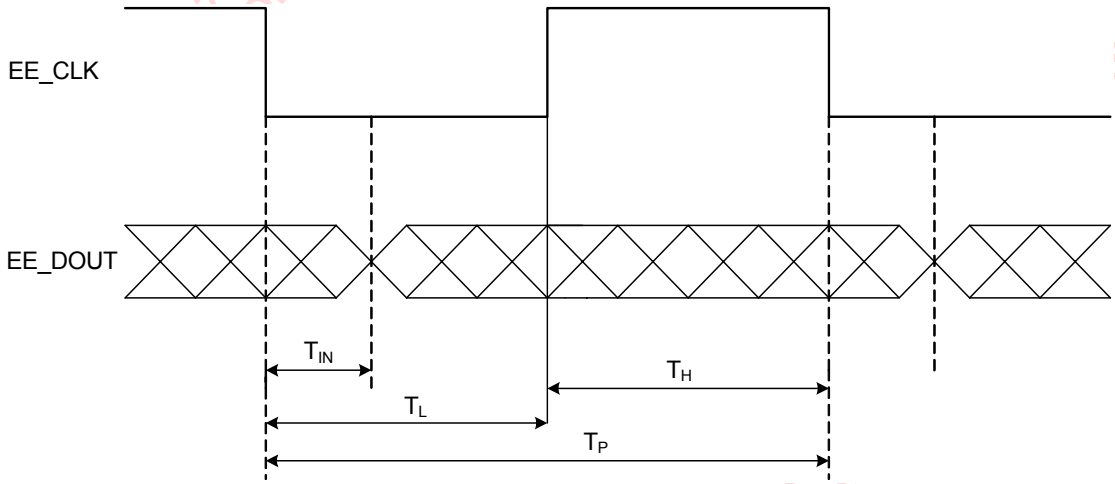
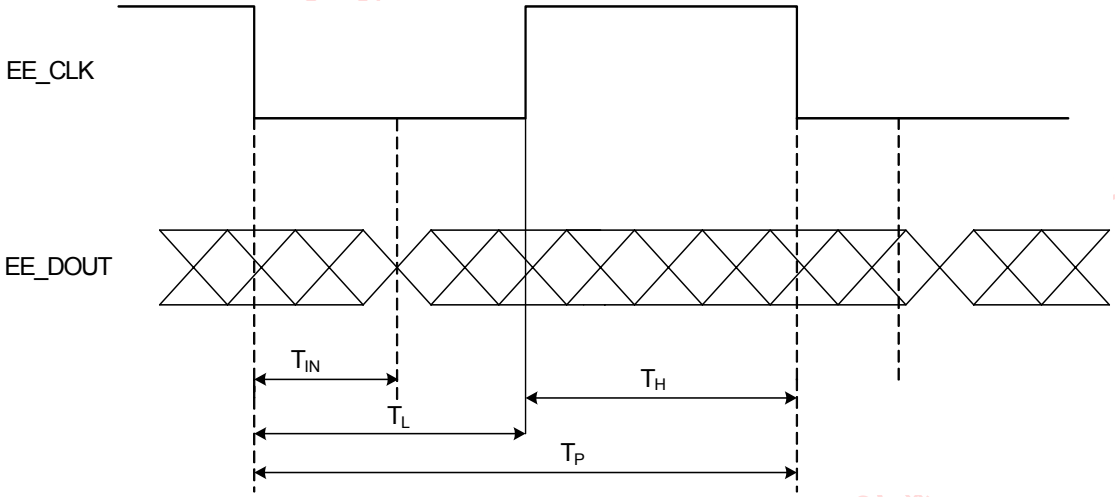


Table 210: 2-Wire EEPROM Output Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units
T_P	EE_CLK period			19200		ns
T_H	EE_CLK high time			9600		ns
T_L	EE_CLK low time			9600		ns
T_{IN}	EE_CLK output time		0		9800	ns

Figure 101:2-Wire Output Timing



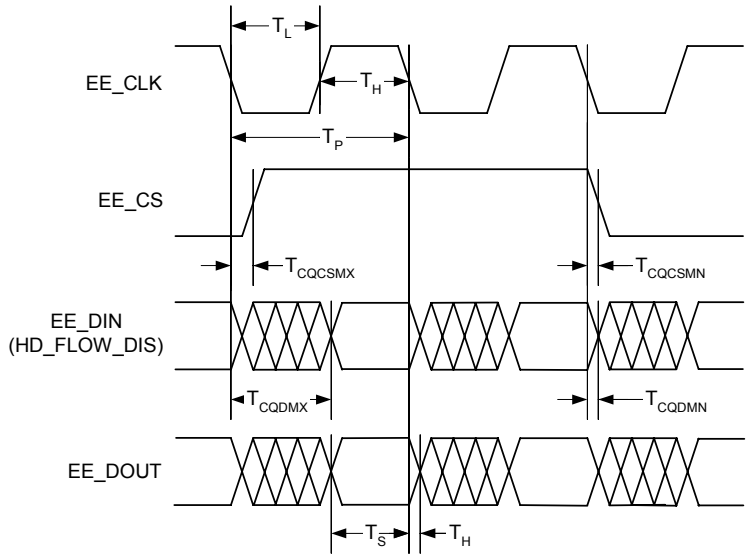
16.12.2 4-Wire EEPROM Timing

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

Table 211: 4-Wire EEPROM Timing

Symbol	Parameter	Condition	Min	Typ	Max	Units	Notes
T_P	EE_CLK period			9600		ns	
T_H	EE_CLK high time			4800		ns	
T_L	EE_CLK low time			4800		ns	
T_{CQCSMX}	Serial EEPROM chip select valid	Referenced to EE_CLK			5	ns	
T_{CQCSMN}	Serial EEPROM chip select invalid				5	ns	
T_{CQDMX}	Serial EEPROM data transmitted to EEPROM valid				10	ns	
T_{CQDMN}	Serial EEPROM data transmitted to EEPROM invalid		3			ns	
T_S	Setup time for data received from EEPROM		10			ns	
T_H	Hold time for data received from EEPROM		10			ns	

Figure 102:4-Wire EEPROM Timing



16.13 SERDES (Serial Interface) Timing

Table 212: SERDES (Serial Interface) Transmitter AC Characteristics

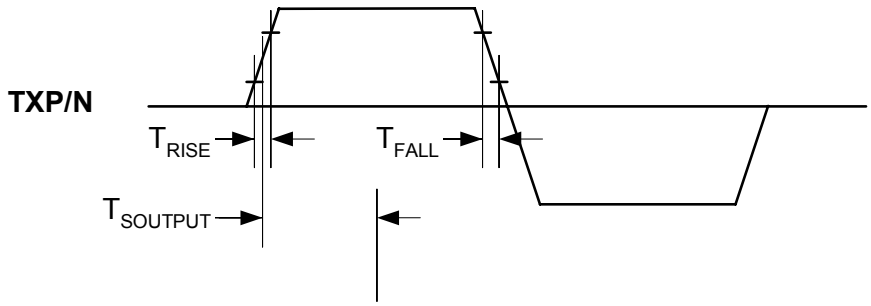
Symbol	Parameter	Min	Typ	Max	Units
CLOCK	Clock signal duty cycle @ 125 MHz	45		55	%
T_{FALL}	V_{OD} Fall time (20% - 80%)	80		200	ps
T_{RISE}	V_{OD} Rise time (20% - 80%)	80		200	ps

Table 213: SERDES (Serial Interface) Receiver AC Characteristics

Symbol	Parameter	Min	Typ	Max	Units
$T_{SOUTPUT}$[1]	SERDES output	360	400	440	ps

1. Measured at 50% of the transition.

Figure 103: SERDES Rise and Fall Times



16.14 IEEE AC Parameters (Ports 0–7)

IEEE tests are typically based on templates and cannot simply be specified by number. For an exact description of the templates and the test conditions, refer to the IEEE specifications:

-10BASE-T IEEE 802.3 Clause 14-2000

-100BASE-TX ANSI X3.263-1995

-1000BASE-T IEEE 802.3ab Clause 40 Section 40.6.1.2 Figure 40-26 shows the template waveforms for transmitter electrical specifications.

(Over full range of values listed in the Recommended Operating Conditions unless otherwise specified)

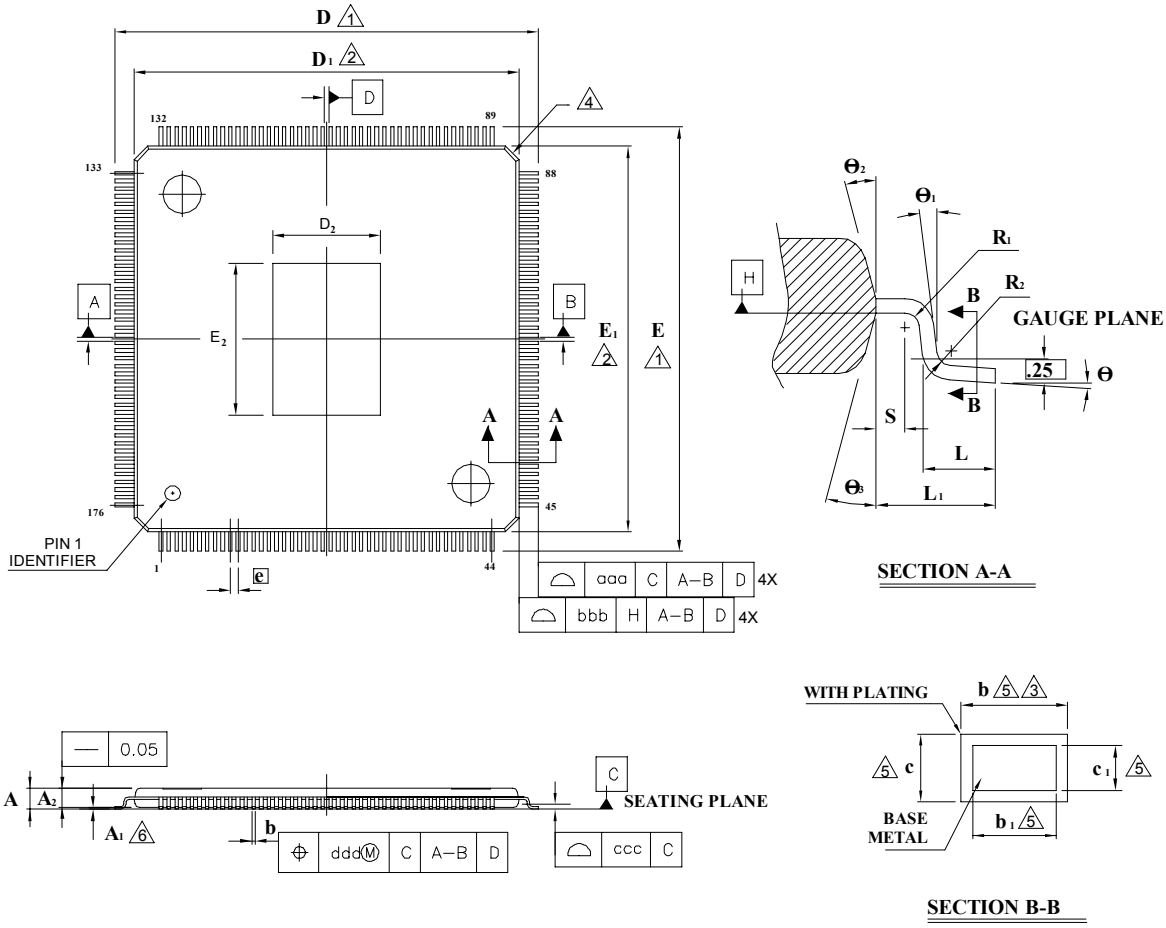
Table 214: IEEE AC Parameters

Symbol	Parameter	Pins	Condition	Min	Typ	Max	Units
T_{RISE}	Rise time	TXP/N[7:0]	100BASE-TX	3.0	4.0	5.0	ns
T_{FALL}	Fall time	TXP/N[7:0]	100BASE-TX	3.0	4.0	5.0	ns
T_{RISE}/ T_{FALL} Symmetry		TXP/N[7:0]	100BASE-TX	0		0.5	ns
DCD	Duty cycle distortion	TXP/N7:0]	100BASE-TX	0		0.5[1]	ns, peak-peak
Transmit Jitter		TXP/N[7:0]	100BASE-TX	0		1.4	ns, peak-peak

1. ANSI X3.263-1995 Table 3.

Section 17. Package Mechanical Dimensions

Figure 104:88E6096/88E6097 176-pin TQFP EPAD Package Mechanical Drawings



NOTE :

1. TO BE DETERMINED AT SEATING PLANE C .
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR CAN NOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.
4. EXACT SHAPE OF EACH CORNER IS OPTIONAL.
5. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10 mm AND 0.25 mm FROM THE LEAD TIP.
6. A1 IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE LOWEST POINT OF THE PACKAGE BODY.
7. CONTROLLING DIMENSION : MILLIMETER.

Table 215: 88E6096/88E6097 176-Pin TQFP EPAD Package Dimensions in mm

Symbol	Dimension in mm		
	Min	Nom	Max
A	1.00	1.10	1.20
A_1	0.05	0.10	0.15
A_2	0.95	1.00	1.05
b	0.13	0.18	0.23
b_1	0.13	0.16	0.19
c	0.09	--	0.20
c_1	0.09	--	0.16
D	22.00 BSC		
D_1	20.00 BSC		
E	22.00 BSC		
E_1	20.00 BSC		
e	0.40 BSC		
L	0.45	0.60	0.75
L_1	1.00 REF		
R_1	0.08	--	--
R_2	0.08	--	0.20
S	0.20	--	--
D_2	7.11		
E_2	7.87		
q	0°	3.5°	7°
θ_1	0°	--	--
θ_2	11°	12°	13°
θ_3	11°	12°	13°
aaa	0.20		
bbb	0.20		
ccc	0.08		
ddd	0.07		

Figure 105:88E6097F 216-pin LQFP EPAD Package Mechanical Drawings

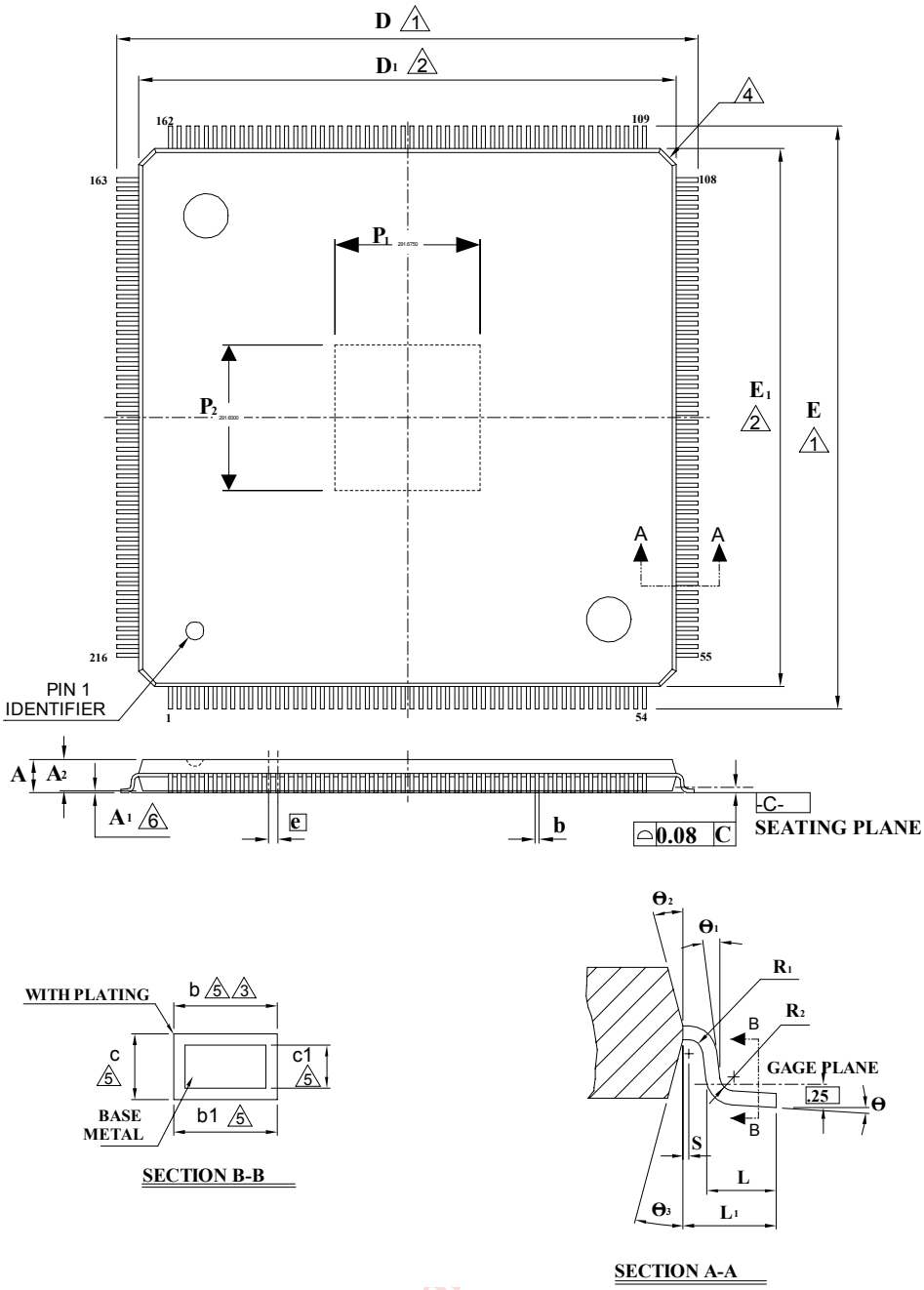


Table 216: 88E6097F 216-pin LQFP EPAD Package Dimensions

Symbol	Dimension in mm		
	Min	Nom	Max
A	--	--	1.60
A_1	0.05	--	0.15
A_2	1.35	1.40	1.45
b	0.13	0.18	0.23
b_1	0.13	0.16	0.19
c	0.09	0.14	0.20
c_1	0.09	0.12	0.16
D	25.60	26.00	26.40
D_1	--	24.00	--
E	25.60	26.00	26.40
E_1	--	24.00	--
e	0.40 BSC		
L	0.45	0.60	0.75
L_1	1.00 REF		
R_1	0.08	--	--
R_2	0.08	--	--
S	0.20	--	--
P_1	7.87		
P_2	7.87		
θ	0°	3.5°	7°
θ_1	0°	--	--
θ_2	11°	12°	13°
θ_3	11°	12°	13°

Notes:

1. TO BE DETERMINED AT SEATING PLANE -C-.
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR CAN NOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.
4. EXACT SHAPE OF EACH CORNER IS OPTIONAL.
5. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10 mm AND 0.25 mm FROM THE LEAD TIP.
6. A1 IS DEFINED AS THE DISTANCE FROM THE SEATING PLANE TO THE LOWEST POINT OF THE PACKAGE BODY.
7. CONTROLLING DIMENSION: MILLIMETER.

Section 18. Ordering Information

18.1 Ordering Part Numbers and Package Markings

Figure 106 shows the ordering part numbering scheme for the devices. Contact Marvell® FAEs or sales representatives for complete ordering information.

Figure 106:Sample Part Number

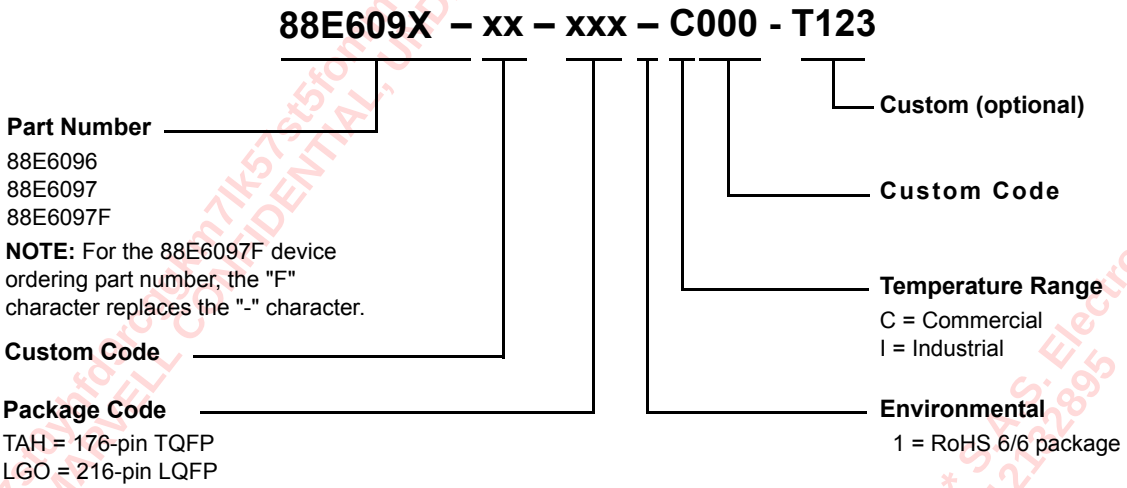


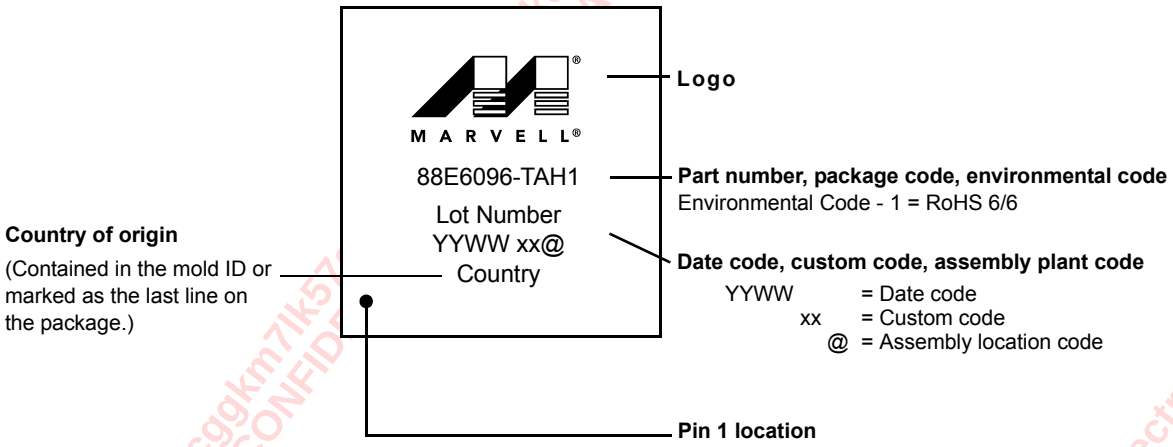
Table 217: 88E6096/88E6097/88E6097F Part Order Options - RoHS 6/6 Compliant Package

Package Type	Part Order Number
88E6096 176-pin TQFP - Commercial	88E6096-XX-TAH1C000
88E6097 176-pin TQFP - Commercial	88E6097-XX-TAH1C000
88E6097 176-pin TQFP - Industrial	88E6097-XX-TAH1I000
88E6097F 216-pin LQFP - Commercial	88E6097FXX-LGO1C000
88E6097F 216-pin LQFP - Industrial	88E6097FXX-LGO1I000

18.1.1 RoHS 6/6 Compliant Marking Examples

Figure 107 is an example of the package marking and pin 1 location for the 88E6096 176-pin TQFP Commercial RoHS 6/6 compliant package.

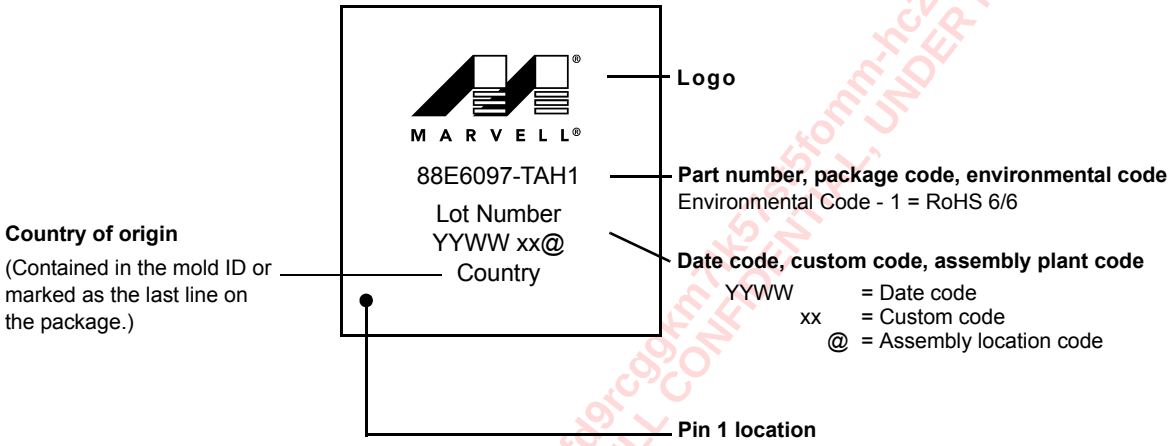
Figure 107:88E6096 176-pin TQFP Commercial RoHS 6/6 Compliant Package Marking and Pin 1 Location



Note: The above example is not drawn to scale. Location of markings is approximate.

Figure 108 is an example of the package marking and pin 1 location for the 88E6097 176-pin TQFP Commercial RoHS 6/6 compliant package.

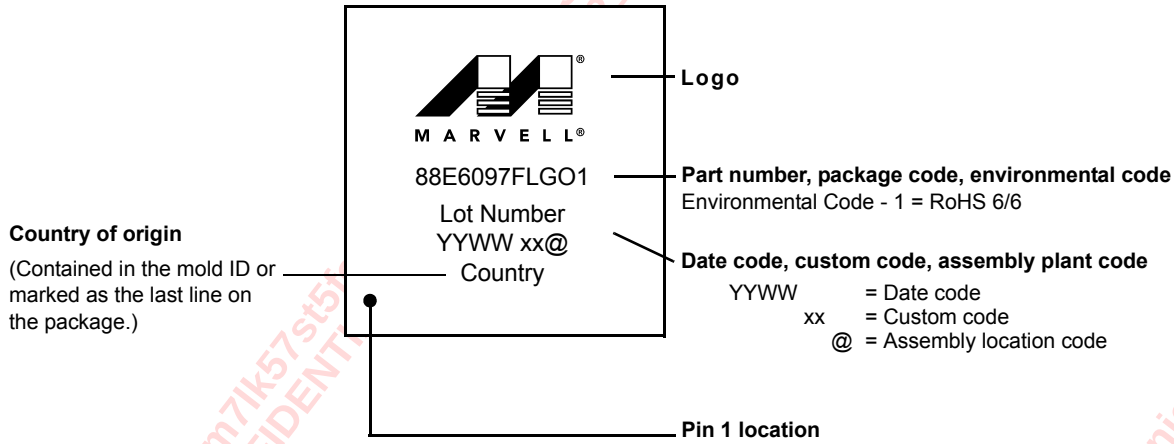
Figure 108:88E6097 176-pin TQFP Commercial RoHS 6/6 Compliant Marking and Pin 1 Location



Note: The above example is not drawn to scale. Location of markings is approximate.

Figure 109 is an example of the package marking and pin 1 location for the 88E6097F 216-pin LQFP Commercial RoHS 6/6 compliant package.

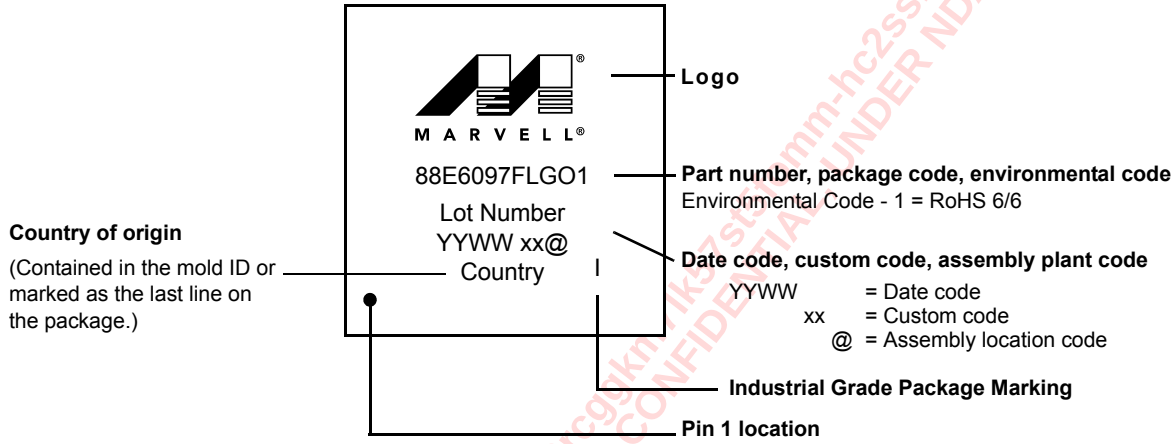
Figure 109:88E6097F 216-pin LQFP Commercial RoHS 6/6 Compliant Package Marking and Pin 1 Location



Note: The above example is not drawn to scale. Location of markings is approximate.

Figure 110 is an example of the package marking and pin 1 location for the 88E6097F 216-pin LQFP Industrial RoHS 6/6 compliant package.

Figure 110:88E6097F 216-pin LQFP Industrial RoHS 6/6 Compliant Package Marking and Pin 1 Location



Note: The above example is not drawn to scale. Location of markings is approximate.

MOVING FORWARD FASTER®

Marvell Semiconductor, Inc.

5488 Marvell Lane
Santa Clara, CA 95054, USA

Tel: 1.408.222.2500
Fax: 1.408.752.9028

www.marvell.com

Worldwide Corporate Offices

Marvell Semiconductor, Inc.
5488 Marvell Lane
Santa Clara, CA 95054, USA
Tel: 1.408.222.2500

Marvell Asia Pte, Ltd.
151 Lorong Chuan, #02-05
New Tech Park, Singapore 556741
Tel: 65.6756.1600
Fax: 65.6756.7600

Marvell Japan K.K.
Shinjuku Center Bldg. 44F
1-25-1, Nishi-Shinjuku, Shinjuku-ku
Tokyo 163-0644, Japan
Tel: 81.(0).3.5324.0355
Fax: 81.(0).3.5324.0354

Marvell Semiconductor Israel, Ltd.
6 Hamada Street
Mordot HaCarmel Industrial Park
Yokneam 20692, Israel
Tel: 972.(0).4.909.1500
Fax: 972.(0).4.909.1501

Marvell Semiconductor Korea, Ltd.
Rm. 603, Trade Center
159-2 Samsung-Dong, Kangnam-Ku
Seoul 135-731, Korea
Tel: 82.(0).2.551-6070/6079
Fax: 82.(0).2.551.6080

Radlan Computer Communications, Ltd.
Atidim Technological Park, Bldg. #4
Tel Aviv 61131, Israel
Tel: 972.(0).3.645.8555
Fax: 972.(0).3.645.8544

Worldwide Sales Offices

Western US
Marvell
5488 Marvell Lane
Santa Clara, CA 95054, USA
Tel: 1.408.222.2500
Fax: 1.408.752.9028
Sales Fax: 1.408.752.9029

Central US
Marvell
9600 North MoPac Drive, Suite #215
Austin, TX 78759, USA
Tel: 1.512.343.0593
Fax: 1.512.340.9970

Eastern US/Canada
Marvell
Parlee Office Park
1 Meeting House Road, Suite 1
Chelmsford, MA 01824 , USA
Tel: 1.978.250.0588
Fax: 1.978.250.0589

Europe
Marvell
5 Marchmont Gate
Boundary Way
Hemel Hempstead
Hertfordshire, HP2 7BF
United Kingdom
Tel: 44.(0).1442.211668
Fax: 44.(0).1442.211543

Israel
Marvell
6 Hamada Street
Mordot HaCarmel Industrial Park
Yokneam 20692, Israel
Tel: 972.(0).4.909.1500
Fax: 972.(0).4.909.1501

China
Marvell
5J1, 1800 Zhongshan West Road
Shanghai, PRC 200233
Tel: 86.21.6440.1350
Fax: 86.21.6440.0799

Marvell
Rm. 1102/1103, Jintian Fudi Mansion
#9 An Ning Zhuang West Rd.
Qing He, Haidian District
Beijing, PRC 100085
Tel: 86.10.8274.3831
Fax: 86.10.8274.3830

Japan
Marvell
Shinjuku Center Bldg. 44F
1-25-1, Nishi-Shinjuku, Shinjuku-ku
Tokyo 163-0644, Japan
Tel: 81.(0).3.5324.0355
Fax: 81.(0).3.5324.0354

Taiwan
Marvell
2Fl., No.1, Alley 20, Lane 407, Sec. 2
Ti-Ding Blvd., Nei Hu District
Taipei, Taiwan, 114, R.O.C
Tel: 886.(0).2.8177.7071
Fax: 886.(0).2.8752.5707

Korea
Marvell
Rm. 603, Trade Center
159-2 Samsung-Dong, Kangnam-Ku
Seoul 135-731, Korea
Tel: 82.(0).2.551-6070/6079
Fax: 82.(0).2.551.6080

For more information, visit our website at:
www.marvell.com