Link Street®
88E6341/88E6141
Switch Functional Specification

Doc. No. MV-S110899-00, Rev. --
November 19, 2015
CONFIDENTIAL
Document Classification: Proprietary Information

Document Conventions

	**Note:** Provides related information or information of special importance.
	**Caution:** Indicates potential damage to hardware or software, or loss of data.
	**Warning:** Indicates a risk of personal injury.

Document Status

Doc Status: Draft	Technical Publication: 0.02

For more information, visit our website at: www.marvell.com

Preface

About this Document

The 88E6341/88E6141 Switch Functional Specification is part of a multi-part set that includes the following documents:

- **88E6341 or 88E6141 Datasheets: Overview, Pinout, Applications, Mechanical and Electrical Specifications**
  Provides a feature list and overview describing the 88E6341 or 88E6141 devices. These datasheets also provide the pin description, pin map, mechanical drawings, and electrical specifications for each device.

- **Link Street® 88E6341/88E6141 PHY and SERDES Functional Specification**
  Provides a functional description of the PHY and SERDES.

- **Link Street® Integrated Management Processor (IMP) Functional Specification**
  Provides a description of the Integrated Management Processor.

Table of Contents

List of Tables

List of Figures

1 Register Description

The device's 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 (see MDC/MDIO in pin list document). The device supports two different kinds of SMI address usage models. One uses 1 of the 32 possible Device addresses (when in multi chip mode – section Multi Chip Addressing Mode). The other uses all of the 32 possible Device addresses (when in single chip mode – section Single Chip Addressing Mode). The device addresses used and mode is configurable at Reset with the ADDR[3:0] configuration pins.

1.1 Register Types

The registers in the device 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.

Table 1: Register Types

Type	Description
LH	Register field with latching high function. If status is high, then the register is set to a 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 is cleared to a zero and remains cleared 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.
ROC	Read only clear. After read, register field is cleared to zero.
RWC	Read/Write clear on read. All bits are readable and writable. After reset or after the register is read, the register field is cleared to zero.
RWR	Read/Write reset. All bits are readable and writable. After reset the register field is cleared to zero.
RWS	Read/Write set. All bits are readable and writable. After reset the 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 cleared to zero when the function is complete.
Update	Value written to the register field does not take effect until a soft reset is executed.
WO	Write only. Reads to this type of register field return undefined data.

1.2 Multi Chip Addressing Mode

When Multi chip addressing mode is used the device responds to only 1 of the 32 possible SMI device addresses so it can share the SMI interface with multiple devices. The SMI address that is used is determined by the ADDR[3:0] configuration pins. ADDR[4] must be zero to select the device. In this mode only two device registers are directly accessible, the SMI Command register (Table 2, SMI Command Register, on page 16) and the SMI Data register (Table 3, SMI Data Register, on page 16). These two registers are used to indirectly access all the other registers (along with any PHY registers that may be attached to it).

Indirect accessing of the other device registers is accomplished by setting the SMI Command register's DevAddr and RegAddr bits to point to the device register to access. Use the DevAddr and RegAddr values defined for the device in the Single Chip Addressing mode (Section 1.3, Single Chip Addressing Mode)

Multi Chip Addressing Mode is enabled when the ADDR[3:0] configuration pins (see pin list document) are a non-zero value at the rising edge of RESETn. The ADDR[3:0] configuration pins also define the single SMI address this device will respond to. This requires that all devices on the same SMI interface use unique ADDR[3:0] values or conflicts will occur. An SMI address of 0x00 is not supported in this mode as this value on the ADDR[3:0] pins places the device into Single Chip Addressing mode (section Single Chip Addressing Mode).

Note: The ADDR[3:0] configuration pins on the device are called ADDR[3:0]n, where the 'n' at the end of the name indicates the physical configuration pins get inverted before being used as ADDR[3:0].

Table 2: 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 completes this bit will 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 = Do not Generate IEEE 802.3 Clause 22 SMI frames 1 = Generate IEEE 802.3 Clause 22 SMI frames
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): 0x0 = Reserved 0x1 = Write Data Register 0x2 = Read Data Register 0x3 = Reserved
9:5	DevAddr	RWR	Internal SMI Device Address bits. These bits are used to select the (Clause 22) SMI device to operate on during SMI commands.
4:0	RegAddr	RWR	Internal SMI Register Address bits. These bits are used to select the (Clause 22) SMI register to operate on during SMI commands.

1. This register is accessible only when the device is in Multi Chip Addressing mode.

Table 3: SMI Data Register[1]
Offset: 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 completes (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.

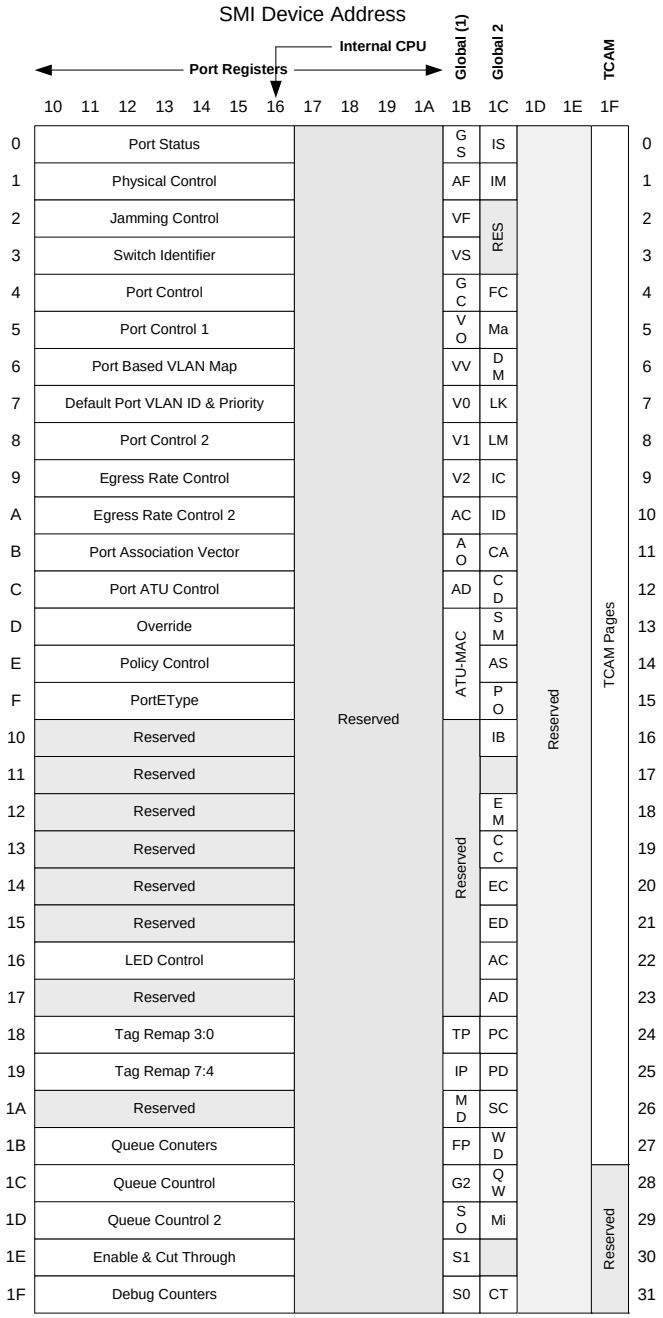
1.3 Single Chip Addressing Mode

Table 4 below, shows the register map assuming the single chip mode is being used. In this mode the device responds to the upper 16 out of the 32 possible SMI device addresses (typically controlled by a CPU). If external PHYs are attached to the device's ports 0 and/or 5 they should use SMI device addresses 0x10 (for Port 0), and 0x15 (for Port 5) so the device can automatically poll the PHYs for link, speed, duplex and flow control status, and communicate the current PHY state to the correct MAC. If the external PHYs cannot support these SMI device addresses then the PHY Mapping Table (Global 2 offsets 0x18 & 0x19) can be configured to map the external PHY's SMI address to the physical port it is connected to. The registers in the external PHYs can be accessed by using this device's SMI PHY Command and Data registers (Global 2 offsets 0x18 & 0x19) using SMIFunc set for External Access mode.

Single Chip Addressing Mode is enabled when the ADDR[3:0]n configuration pins (see pin list document) are all high at the rising edge of RESETn.

Notes: The internal PHY and SERDES registers are only accessible via the SMI PHY Command and Data registers (Global 2 offsets 0x18 & 0x19) using SMIFunc set for Internal Access mode. The internal Port 1 to Port 4 PHYs are mapped at SMI device addresses 0x11 to 0x14 respectively. The internal SERDES on Port 5 is mapped at SMI device addresses 0x15.

Table 4: 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
IP = IP PRI Mapping Table
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
FC = Flow Control Delays
Ma = Management
DM = Device Mapping
LK = LAG Mask

LM = LAG 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
IB = I2C Block
EM = Energy Management
CC = CPU Communications
EC & ED = EEPROM Cmd & Data
AC & AD = AVB Cmd & Data

PC = SMI PHY Command
PD = SMI PHY Data
SC = Scratch
WD = Watch Dog Control
QW = QoS Weights
Mi = Misc
CT = Cut Through Control

1.3.1 Switch Port Registers

Each Ethernet port in the device contains its own per port registers. Each per port register is 16-bits wide and their bit assignments are shown in Table 5.

Table 5: Per Port Register bit Map

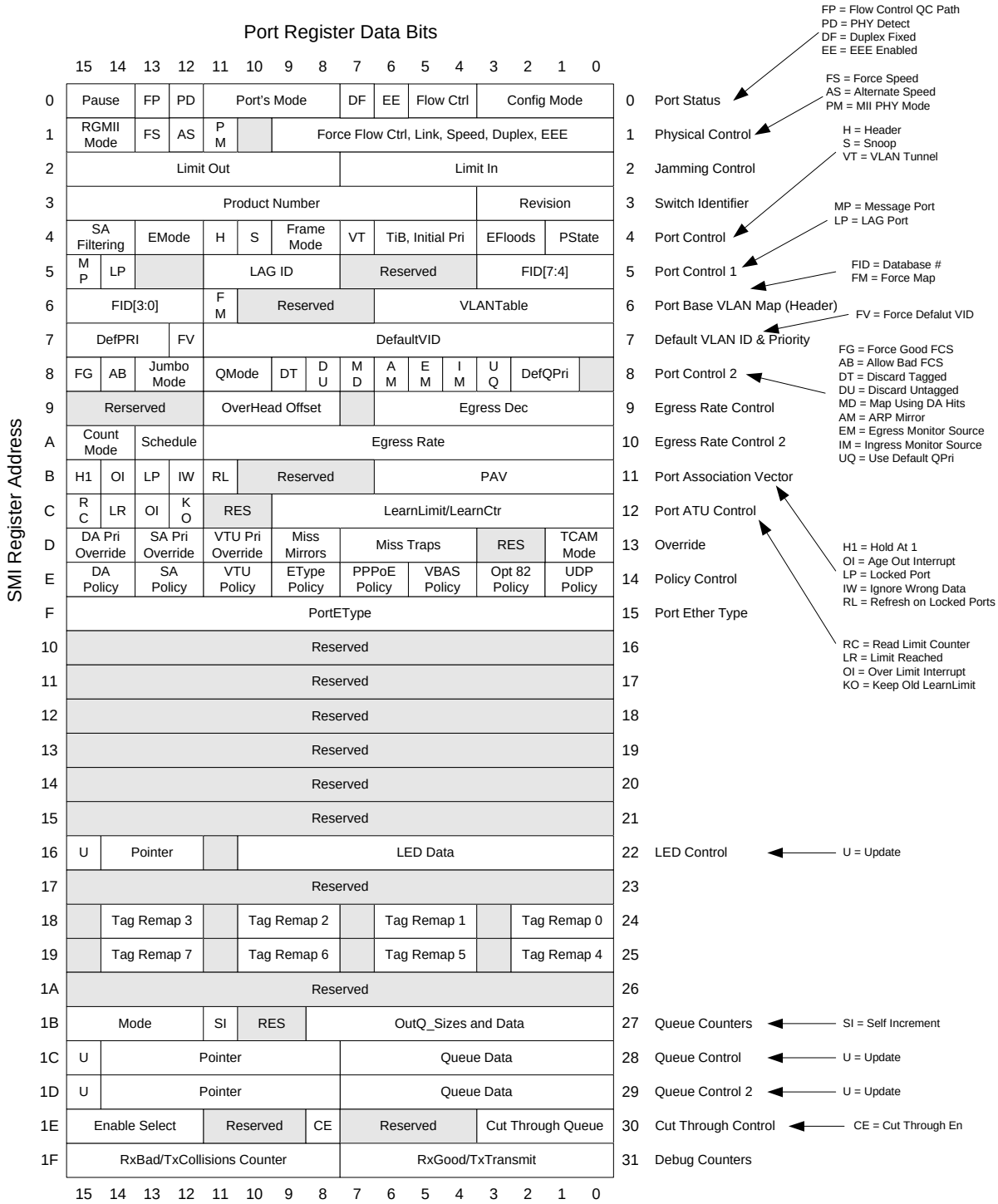


Table 6: Port Status Register[1]
Offset: 0x00 or decimal 0 - Port

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, below, is a one. This bit reflects the Pause result from auto-negotiation only (i.e., the ForceFC 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 & in MyPause
14	MyPause	RO	My Pause bit. My Pause bit sent to the PHY during PHY Polling Unit (PPU) initialization. This bit is not meaningful on ports that don't support Auto-Negotiation (i.e., Internal ports). It is set high if FLOW (see the relevant product's datasheet pin list for the FLOW pin details) is high during RESETn. 0 = MAC Pause is not to be advertised as supported in this port and Half-duplex back pressure will not be used on this port. 1 = MAC Pause is to be advertised as supported in this port and Half-duplex back pressure will be used on this port if this port is in half-duplex mode. This bit reflects the Reset value of the FLOW pin. Its value is not changed by forcing flow control on or off on this port.
13	FCPath	RWR	Flow Control QC Path Considerations. When this bit is cleared to a zero, the QC will use its Flow Control algorithm only when both the source port of a frame and its destination port have Flow Control enabled (the source of this information includes the effect of the TagFlowControl bit – Global 2 offset 0x05). This mode is intended for external network ports and cross-chip DSA ports. When this bit is set to a one, the QC will use its Flow Control algorithm whenever the source port of a frame has Flow Control enabled (the source of this information includes the effect of the TagFlowControl bit – Global 2 offset 0x05). This mode is intended for ports that are connected to internal devices that need local flow control.
12	PHYDetect This bit has no effect on the internal CPU's Port – a value of 0x0 is used.	RWR	802.3 PHY Detected. This bit is set to a one if the device is enabled and it finds a non all one's value in either SMI registers 2 or 3 at the same SMI device address as this Port's address when the PHY is accessed using Global 2 offsets 0x18 & 0x19. 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 device's init routine (that is why this register must not be written to while the InitState, Global 1 offset 0x00, is Initializing – i.e., 0x0). The device uses these bits to determine which port's to poll the PHYs for Link, Duplex, Speed and Flow Control. If software changes these bits the device will change its polling accordingly.

Table 6: Port Status Register[1] (Continued)
Offset: 0x00 or decimal 0 - Port

Bits	Field	Type	Description
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 comes from the LinkValue bit if the link is being forced by ForcedLink being a one (Port offset 0x01). Otherwise the port's Link bit comes from the source defined in Table 7, Interface Configuration Matrix, on page 23. For Port 0 (the xMII interface) the port's Px_ENABLE represents the link status. The Px_ENABLE comes from any available GPIO pin – see Enable Select (Port offset 0x1F). Link is up on the internal CPU's switch port (Port 0x1E) whenever the internal CPU's Rx NIC is enabled (RxDMA at I/O 0x20) AND its Tx NIC is enabled (TxDMA at I/O 0x28), or when the Switch Loop Back bit is set (Global 2 offset 0x013 index 0x0F).
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 comes from the DpxValue bit if the duplex is being forced by ForcedDpx being a one (Port offset 0x01). Otherwise the port's Duplex bit comes from the source defined in Table 7 on page 23.
9:8	Speed	RO	Speed mode. These bits are valid when the Link bit, above, is set to a one. 0x0 = 10 Mbps 0x1 = 100 or 200 Mbps 0x2 = 1000 Mbps 0x3 = 2500 Mbps The port's Speed bits comes from the ForceSpeed bits if the speed is being forced by ForceSpeed being non-0x3 (Port offset 0x01). Otherwise the port's Speed bits come from the source defined in Table 7, Interface Configuration Matrix, on page 23. **NOTE:** These bits will reflect the actual speed of Ports 0 and 5 only when an external PHY is attached to the port. Otherwise these bits will reflect the C_Mode speed of the port (as best it can) if the speed is not being forced.

Table 6: Port Status Register[1] (Continued)
Offset: 0x00 or decimal 0 - Port

Bits	Field	Type	Description
7	Duplex Fixed	RO	Duplex Fixed. When this bit is set to a one a Duplex Mismatch was detected on this port and its duplex has been AutoFixed (If AutoFixMismatch is enabled in Global 2 offset 0x1B). It will clear to zero when link from the MAC on this port is detected or forced down. This bit being a one means the port is being forced into full duplex mode – but this won't be seen in the port's forcing register bits as the forcing register bits can still be used to override this force.
6	EEE Enabled	RO	EEE (Energy Efficient Ethernet) Enabled from the PHY. This bit is set to a one when the PHY on this port is advertising support for EEE and its link partner also advertises support for EEE. This bit only valid on ports with integrated PHYs. This bill will always be zero on the MII ports. This bit is the ANDing of the two signals from the PHY (local and link partner EEE advertise bits). This bit is not affected If EEE is forced on or off (Port offset 0x01).
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 will be used by the Tx MAC (i.e., the transmitter will be paused off). If the port is in half duplex mode this bit will never be a one since all Rx Pause frames will be ignored. If the port is in full duplex mode this bit will never be 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 being a one indicates that the port is or will be using back pressure. If the port is in full duplex mode this bit 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.
3:0	C_Mode (not all modes available on all ports)	RO	Config Mode. These bits return the switch port's current interface type configuration mode. Most of these modes are determined at reset based on the physical design of the device and/or the port's Px_MODE configuration (CONFIG) bit values at reset. Basically, a port's C_Mode bits equal the port's Px_MODE (for ports that have Px_MODE configuration pins) after reset.

1. This PortStatus register must not be written to if the InitState in the Global Status register (section Switch Global Status Register (Table 58 p. 87)) is 0x0.

The source of the port's Link, Speed and Duplex bits for each possible C_Mode is defined in Table 7, Interface Configuration Matrix, on page 23 (assuming no forcing of the bits is occurring – Port offset 0x01). Each of these bits, Link, Speed and Duplex, can be individually forced to any value by using the Forcebits in the Physical Control register (Port offset 0x01).

Not all C_Mode values are supported by all the ports. The list of ports that support each C_Mode is noted under the Switch Port's column in Table 7 on page 23.

Table 7: Interface Configuration Matrix

C_Mode	PHY Detect	Name	Switch Port(s)	Clock Mode[1]	Link[2]	Speed[3]	Duplex[4]
0x0	X	FD MII[5]	0	Input (can be Output)	Px_ENABLE	100	FD Only
0x1	X	MII PHY	0	Output 2.5, 25 or 50 MHz	Px_ENABLE	100	FD
0x2	0	MII MAC	0	Input	Px_ENABLE	Link Partner[6]	FD
	1	MII to PHY	0	Input	PPU	PPU	PPU
0x3	X	GMII	6	Embedded	Internal CPU[7]	1000	FD Only
0x4	0	RMII PHY	0	Output 50 MHz	Px_ENABLE	100	FD
	1	RMII to PHY	0	Output 50 MHz	PPU	PPU	PPU
0x5	0	RMII MAC	0	Input	Px_ENABLE	100	FD
	1	RMII to PHY	0	Input	PPU	PPU	PPU
0x6	X	xMII Tristate	0	Off – TriState	0	X	X
0x7	0	RGMII	0	Source Sync	Px_ENABLE	1000	FD
	1	RGMII to PHY	0	Source Sync	PPU	PPU	PPU
C_Mode	**PHY Detect**	**Name**	**Switch Ports**	**Clock Mode**	**Link**	**Speed**	**Duplex**
0x8	-	Reserved	-	-	-	-	-
0x9	0	1000BASE-X	5	Embedded	SERDES	1000	FD Only
0xA	1	SGMII	5	Embedded	SERDES & PPU	PPU	PPU
0xB	X	2500BASE-X	5	Embedded	SERDES	2500	FD Only
0xC	X	Reserved	-	-	-	-	-
0xD	X	Reserved	-	-	-	-	-
0xE	-	Reserved	-	-	-	-	-
0xF	1	PHY	1 to 4	From PHY	PHY	PHY	PHY

1. The digital xMII's are Tristated if their Link goes down by the Px_ENABLE pin (see Port offset 0x1E).
2. The digital xMII's are Tristated if their Link goes down by the Px_ENABLE pin (see Port offset 0x1E).
3. C_Modes 0x0, 0x1, 0x4 and 0x5 on Port 0 default to a Speed of 100 but they can be forced to 10 Mbit operation (see Port offset 0x01). For C_Mode 0x1 and C_Mode 0x0, when it is in PHY mode, the Clock Mode's frequency will also change accordingly.
4. All digital xMII's mode defaults to full duplex operation, but 10 or 100 speed modes can be forced into half duplex operation (see Port offset 0x01) with the exception of C_Mode 0x0 which can support full duplex only.
5. C_Mode 0x0 starts out as MAC mode where INCLK and OUTCLK are inputs. But the interface can be reconfigured to PHY mode where INCLK and OUTCLK are outputs (see Port offset 0x01).
6. The Link Partner's Input clocks determine the actual speed of the MII. The Speed bits (see Port offset 0x00) will not reflect the actual speed of the port unless software updates the ForceSpd bits (Port offset 0x01).
7. Link is up on the internal CPU's switch port (Port 0x16) whenever the internal CPU's Rx NIC is enabled (RxDMA at I/O 0x20) AND its Tx NIC is enabled (xDMA at I/O 0x28) or when Switch Loop back is on (Global 2 offset 0x013 index 0x0F).

Note: C_Mode is initially set to the port's Px_MODE (see Px_OUTD pin description).

Note: The PPU is the PHY Polling Unit that automatically scans external PHYs for their status using this device's MDC_PHY and MDIO_PHY pins assuming the port's PHYDetect bit is set (Port offset 0x00).

Table 8: Physical Control Register
Offset: 0x01 or decimal 1 – Port

Bits	Field	Type	Description
15	RGMII Rx Timing (valid on Port 0 only)	RWR	RGMII Receive Timing Control. Changes to this bit are disruptive to normal operation. Hence any change to this register must be done only while the port's link is down (see bits 5:4 below). 0 = Default 1 = Add delay to RXCLK for IND inputs when port is in RGMII mode See relevant product's datasheet for RGMII Timing Mode details.
14	RGMII Tx Timing (valid on Port 0 only)	RWR	RGMII Transmit Timing Control. Changes to this bit are disruptive to normal operation. Hence any change to this register must be done only while the port's link is down (see bits 5:4 below). 0 = Default 1 = Add delay to GTXCLK for OUTD outputs when port is in RGMII mode See Section <RGMII Timing> for RGMII Timing Modes.
13	ForcedSpd	RWR	Force Speed. When this bit is set to a one the speed for this port's MAC will be forced to the value in the SpdValue and the AltSpeed registers (below) regardless of what the normal speed value would be. When this bit is cleared to a zero, normal speed detection occurs. **NOTE:** Only change the port's speed when its link is down.
12	AltSpeed (valid on Ports 0 & 5 only)	RWR	Alternate Speed Mode. 0 = Normal 10, 100, or 1000 speed (where possible) 1 = Alternate speed as defined in the SpdValue register (below) When Port 0's C_Mode is 0x0 or 0x1 (Port offset 0x00) or when Port 5's C_Mode is 0x9 or 0xB, this bit can be used along with the SpdValue bits below to force the port's speed as listed (if the ForcedSpd bit above is set to a one). **NOTE:** This bit has no effect unless the ForcedSpd bit (above) is set to a one.
11	MII PHY (valid on Port 0 only)	RWR	MII PHY Mode. When Port 0's C_Mode is 0x0 (Port offset 0x00) this bit can be used to configure the port to MAC or PHY mode as follows: 0 = MII MAC mode (Px_INCLK and Px_OUTCLK are inputs) 1 = MII PHY mode (Px_INCLK and Px_OUTCLK are outputs)
10	Reserved	RES	Reserved for future use.

Table 8: Physical Control Register (Continued)
Offset: 0x01 or decimal 1 – Port

Bits	Field	Type	Description
9	EEEValue	RWR	Energy Efficient Ethernet force value. This bit is used to force low power idle mode to be enabled or disabled in the switch MAC's when the ForceEEE bit (below) is set to a one. These bits do not affect the PHY's advertisement. They enabled or disable the MAC's ability to communicate EEE signaling to the PHY. Low power idle mode will be forced enabled when this bit is set to a one. It will be forced disabled when this bit is cleared to a zero. If the ForceEEE bit (below) is cleared to a zero this bit has no effect.
8	ForceEEE	RWR	Force Energy Efficient Ethernet. When this bit is set to a one the low power idle value for this port will be forced to the value in the EEEValue register (above) regardless of what the normal EEE value would be. In this way the low power idle function can be forced to be enabled or disabled. When this bit is cleared to a zero, normal EEE enabling by PHY auto-negotiation occurs.
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 will be forced enabled when this bit is set to a one. It will be 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 0x02) 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 – unless the write is intended to re-enable Pausing on this Jammed port.
6	ForcedFC	RWR	Force Flow Control. When this bit is set to a one flow control (if full duplex) or backpressure (if half duplex) for this port will be 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 0x02) 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 – unless the write is intended to re-enable Pausing on this Jammed 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.

Table 8: Physical Control Register (Continued)
Offset: 0x01 or decimal 1 – Port

Bits	Field	Type	Description
4	ForcedLink	RWR	Force Link. When this bit is set to a one the link for this port's MAC will be 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 down or up. When this bit is cleared to a zero, normal link detection occurs. This bit forces the MAC to consider Link to be up or down. It does not force Link up or down in the PHY, but it will on the MII pins. If software needs to stop the flow of frames on a port it is better to use the Disabled port state in the Port State bits in the Port Control registers (offset 0x04).
3	DpxValue	RWR	Duplex's Forced value. This bit is used to force the duplex to full or half when the ForcedDpx bit (below) is set to a one. The duplex will be 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 (200BASE and faster modes do not support half duplex as undefined results will occur – do not try to force this mode). If the ForcedDpx bit (below) is cleared to a zero this bit has no effect. **NOTE:** The internal CPU port's duplex cannot be changed by changing these bits. It is always full-duplex.
2	ForcedDpx	RWR	Force Duplex. When this bit is set to a one the duplex for this port's MAC 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.

Table 8: Physical Control Register (Continued)
Offset: 0x01 or decimal 1 – Port

Bits	Field	Type	Description
1:0	SpdValue	RWS to 0x3	Speed's Forced value. These bits are used to force the speed on this port's MAC as follows (if the ForcedSpd bit above is set to a one): 0x0 = 10 Mbps 0x1 = 100 or 200 Mbps (depending upon the AltSpeed bit above) 0x2 = 1000 Mbps 0x3 = 2500 Mbps Port 0 is the only port that can support 200 Mbps (if its C_Mode = 0x0 or 0x1 in Port offset 0x00). Port 5 is the only port that can support 2500 Mbps. **NOTE:** The internal CPU port's speed cannot be changed by changing these bits. It is always 1000. **NOTE:** Only change the port's speed when its link is down. These bits change the speed of the port's MAC interface only (and its pins on ports where the port's MAC interface is exposed as a xMII interface[1]). If this port is connected to a PHY and the PHY is not at the same speed as the MAC unpredictable results will occur. Forcing a port to a speed faster than what it can support will set the port to the fastest speed it can support. **NOTE:** This bit has no effect unless the ForcedSpd bit (above) is set to a one.

1. The kind of interface will determine how its speed changes. If the interface's C_Mode (Port offset 0x00) is RGMII and its speed is force to 100 Mbps it will use the RGMII's version of 100 Mbps. If the interface's C_Mode is MII then forcing the port's speed to 1000 Mbps will have no effect.

Table 9: Jamming Control Register
Offset: 0x02 or decimal 2 – Port

Bits	Field	Type	Description
15:8	LimitOut	RWS To 0xFF	Limit Output Pause frames. Limit the amount of time Pause refresh frames transmitted from this port. 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. Clearing 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 Pause refresh frames to egress from this port for a maximum of 1 x 65536 slot times for each congestion situation. Setting these bits to 0x02 will allow Pause refresh frames to egress from this port for a maximum of 2 x 65536 slot times for each congestion situation, etc. The upper 3-bits of this register are a 2^n multiplier for the lower 5 bits. The maximum count is 2^7 * 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.
7:0	LimitIn	RWR	Limit Input Pause frames. 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. 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 10: Switch Identifier Register
Offset: 0x03 or decimal 3 – Port

Bits	Field	Type	Description
15:4	Product Num	RO	Product Number or identifier. The product number is 0x141 or 0x341
3:0	Rev	RO	Revision Identifier. The initial version of the devices has a Rev of 0x0 (Rev 0x0 of the device). This Rev field may change at any time. Please contact Marvell's FAE's for current information on the device revision identifier.

Table 11: Port Control Register
Offset: 0x04 or decimal 4 – Port

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: **0x0 = SA Filtering Disabled** – no frame will be filtered (i.e., discarded) due to the contents of its Source Address field **0x1 = Drop On Lock -** Ingressing frames will be discarded if their SA field is not in the ATU's address database (i.e., it's 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. **0x2 = 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. **0x3 = 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., it's 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 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 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

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 = 0x0 (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 and the Egress TCAM Actions did not define an override. The default modes are: 0x0 = Default to Unmodified mode – frames are transmitted unmodified 0x1 = Default to Transmit all frames Untagged – remove the tag from any tagged frame 0x2 = Default to Transmit all frames Tagged – add a tag to any untagged frame (the IEEE standard Ether Type of 0x8100 is used). 0x3 = Reserved for future use **When Frame Mode = 0x1 (DSA Tag Frames)** these bits must remain at 00 as all other modes are 'Reserved for future use' **When Frame Mode = 0x2 (Provider Tag Frames)** these bits must remain at 00 as all other modes are 'Reserved for future use' **When Frame Mode = 0x3 (Ether Type DSA Tag Frames)** these bits define how non-Control frames (i.e., Forward DSA Tag frames) will look when they egress this port. All non-Control (i.e., non-MGMT) frames will egress as Normal network frames with the following options: 0x0 = Egress Forward DSA frames as Unmodified Normal Network Frames 0x1 = Egress Forward DSA frames as Untagged Normal Network Frames 0x2 = Egress Forward DSA frames as Tagged Normal Network Frames 0x3 = Reserved
11	Header On the internal CPU's Port – a value of 0x1 is expected but not required	RWR	Ingress & Egress Header Mode. When this bit is set to a one all frames Egressing out this port are pre-pended with the Marvell 2 byte Egress Header just before the frame's DA field. Also, all frames Ingressing into this port are expected to be pre-pended with the Marvell 2 byte Ingress Header just before the frame's DA field. On Ingress the 1st 2 bytes after the SFD (Start of Frame Delimiter) 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 (Port 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 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

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[1] overriding the destination port(s) determined by the DA mapping[2]. 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 only (see Frame Mode bits below) and only if Cut Through is disabled on the port (Port offset 0x1F) as the IPv6 Snoop point may be after byte 64.
9:8	Frame Mode On the internal CPU's Port – a value of 0x3 is expected but not required	RWR	Frame Mode. These bits are used to define the expected Ingress and the generated Egress tagging frame format for this port as follows: 0x0 = Normal Network 0x1 = DSA (Distributed Switch Architecture) 0x2 = Provider 0x3 = Ether Type DSA Frame Mode can be overridden by an Egress TCAM Action. **0x0 = 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. **0x1 = 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 LAGs, Mirrors, etc. Ports that are interconnected together to form a larger switch and ports connected to the management CPU must use this mode. **0x2 = 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 device, 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 register will be considered tagged (for the DiscardTag and DiscardUntagged discarding policy – Port offset 0x08), will have the tag's VID and PCP 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 PCP 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 the DiscardTag and DiscardUntagged discarding policy – Port offset 0x08). The ingressing frames are modified so they are ready to egress out Customer ports (Normal Network Frame Mode ports) unmodified.

Table 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

Bits	Field	Type	Description
9:8 (cont.)	Frame Mode On the internal CPU's Port – a value of 0x3 is expected but not required	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 PCP 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). Both the VID and the PCP/DEI bits can be overridden by an Egress TCAM Action. **0x3 = 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 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 (Port offset 0x06), 802.1Q VLANs defined in the VTU (if 802.1Q is enabled – Qmode in Port offset 0x08 and the VTU in Global 1 offsets 0x02 to 0x09) and Trunk Masking (Global 2 offset 0x07) 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 address that is currently 'static' in the ATU. This includes unicast as well as multicast frames. **NOTE:** Do not set this bit to a 0x1 if the port's AvbTunnel bit (AVB Port offset 0x00) is set to a 0x1.

Table 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

Bits	Field	Type	Description
6	TagIfBoth	RWS	Tag if Both. 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 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

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)[3]. 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 frame type overrides (Global 2, offset 0x0F) and/or content overrides if enabled on the port (see Port offset 0x0D, but Fpri content 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 in the Default VLAN ID and Priority register at Port 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: **0x0 = Use Port defaults** for Fpri and Qpri. **0x1 = Use Tag Priority.** If the frame is tagged[4], Fpri is set to the frame's tag PRI bits re-mapped by the port's Tag Remap registers (Port offsets 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. **0x2 = 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). Fpri[2:1] is the frame's Qpri and Fpri[0] is the port's DefFPri[0] unless UseIpFPri is set to a one (Global 1 offset 0x19), in which case Fpri is set by mapping the frame's DiffServ or Traffic Class bits into the IP Mapping Table (Global 1 offset 0x19). If the frame is not Ipv4 nor Ipv6 the port defaults are used for Fpri and Qpri. **0x3 = 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 (Port offsets 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 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

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 is not found, it is considered unknown. Frames with known DA's are not affected 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: 0x0 = Do not egress any frame with an unknown DA (unicast or multicast) 0x1 = Do not egress any frame with an unknown multicast DA 0x2 = Do not egress any frame with an unknown unicast DA 0x3 = 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.

Table 11: Port Control Register (Continued)
Offset: 0x04 or decimal 4 – Port

Bits	Field	Type	Description
1:0	PortState	RWR Or RWS to 0x3[5] Always RWR for the internal CPU's Port	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 or other 802.1 protocols. The state of these bits can be changed at any time without disrupting frames currently in transit. The Port States are: **0x0 = Disabled.** The switch port is completely disabled and it will not receive or transmit any frames. The QC will return any pre-allocated ingress queue buffers when the port is in this mode. The Disabled port state overrides all other PortStates as no buffers are made available for a Disabled port. **0x1 = Blocking/Listening.** The switch will examine all frames without learning any SA addresses, and will discard all but MGMT[6] frames. It will allow MGMT frames only to exit the port. This mode is used for BPDU handling for bridge loop detection and Spanning Tree support or other 802.1 protocols. **0x2 = Learning.** The switch will examine all frames, learning all SA addresses (except those from MGMT frames), and still discard all but MGMT frames. It will allow MGMT frames only to exit the port. **0x3 = Forwarding.** The switch will examine all frames, learning all SA addresses (except those from MGMT frames), and receive and transmit all frames like a normal switch. **NOTE:** The Blocking/Listening, Learning and Forwarding port states can be overridden by per VLAN Port States (see the STU in Global 2 offset 0x05). **NOTE:** The Port State can be automatically configured to the Disabled Port State on any Link down on the port – See Auto Disable in Global 2, offset 0x05.

1. The CPU port is determined by the CPUDest bits in Monitor Control Register (Global offset 0x1A).
2. IGMP/MLD frames that are ingress filtered will not be sent to the CPUDest port.
3. Frames that egress unmodified do not get modified so the frame's tag PRI bits will not be modified.
4. Priority Only tagged frames (tagged frames with a VID=0x000) are considered tagged for priority decisions.
5. The PortState bits for all ports come up in the Forwarding state unless the NO_CPU configuration pin is a zero after Reset, the CPU attached mode, in which case the ports come up Disabled and all internal PHYs & SERDES come up powered down. Ports 0 and 1 always come up in the Disabled port state in the small package regardless of the state of the NO_CPU configuration pin (software can subsequently change their Port State value).
6. MGMT (management) frames are the only kind of frame that can be tunneled through Blocked ports. A MGMT frame is any frame whose DA address appears in the ATU Database with the MGMT Entry State or whose DA is an enabled special multicast (see Rsvd2CPU & Rsbd3AltCpu in Global 2 offset 0x05).

Table 12: Port Control 1
Offset: 0x05 or decimal 5 – Port

Bits	Field	Type	Description
15	Message Port	RWR	Message Port. When the Learn2All bit in the ATU is set to a one (Global 1 offset 0x0A) learning message frames are 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 Marvell Tag ports, except for the CPU's port, have their MessagePort bit set to a one. Ports that are not Marvell Tag ports (i.e., normal Network ports) should not have their MessagePort bit set to a one.
14	LAG Port	RWR	LAG Port. When this bit is set to a one this port is considered to be a member of a LAG (Link Aggregation) with the LAG ID defined below. When this bit is a zero the port is treated as an individual port.
13:12	Reserved	RES	Reserved for future use.
11:8	LAG ID	RWR	LAG ID. When the LAG Port bit (above) is set to a one these bits define which LAG (Link Aggregation) this port is to be associated with. All ports that are members of the same LAG must be assigned the same LAG ID and each group of ports that form a trunk must be assigned unique LAG IDs.
7:4	Reserved	RES	Reserved for future use.
3:0	FID [7:4]	RWR	Port's Default Filtering Information Database (FID) Number bits 7:4. 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 being used. It needs to be a unique number for each independent, non-overlapping, address database if used. The lower 4 bits of the port's default FID are contained in the Port Base VLAN Map register (Port offset 0x06).

Table 13: Port Based VLAN Map
Offset: 0x06 or decimal 6[1] Port

Bits	Field	Type	Description
15:12	FID [3:0]	RWR	Port's Default Filtering Information Database (FID) Number 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, address database if used. The upper 4 bits of the port's default FID are contained in the Port Control 1 register (Port offset 0x05).
11	ForceMap	RWR	Force Mapping. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one all received frames will be considered MGMT[2] and they are mapped to the port or ports defined in the VLANTable bits below overriding the mapping from the address database. The forcing function is needed to get BPDU frames to egress specific ports by the CPU for the Spanning Tree Protocol. ForceMapped frames will egress ports that are not in the Disabled port state (i.e., they are MGMT frames and will egress out Blocked ports). This bit is accessible by the CPU's Ingress Header so the CPU can enable and disable MGMT and forcing on a frame by frame basis. **NOTE:** Learning is disabled on MGMT frames, so this bit being set to a one, also disables learning on frames entering this port.
10:7	Reserved	RES	Reserved for future use.

Table 13: Port Based VLAN Map (Continued)
Offset: 0x06 or decimal 6[1] Port

Bits	Field	Type	Description
6:0	VLANTable	RWS to all ones except for this port's bit	In Chip 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[3], even if 802.1Q is enabled on this port (Port offset 0x08). These bits restrict where a port can send frames to (unless a VLANTunnel frame is being received – Port offset 0x04) or an AVBTunnel frame is being received (AVB Port offset 0x00). If ForceMap is set to a one, these bits indicate which port or ports all frames that ingress this port are sent to overriding the mapping from the address database. 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 will always be zero after Reset. This prevents frames going out the port they came in on. The device allows this Port's bit to be set to a one allowing frames to be switched back to the port they came in on so software needs to be careful with these bits. This register is normally resets to 0x7E for Port0 (SMI Device Address 0x10), and it resets to 0x7D for Port1 (Addr 0x11), to 0x7B for Port2 (Addr 0x12), etc. Cross Chip Port base VLANs are supported by the Cross Chip Port VLAN Table (Global 2 offsets 0x0B and 0x0C).

1. The contents of this register can be modified on a frame by frame basis if the port's Header Mode is enabled (Port offset 0x04). Note: Only the lower 4-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.
2. MGMT = Management frames, frames that can tunnel through Blocked ports (see PortStates in Port offset 0x04).
3. See section (on MGMT frames).

Table 14: Default Port VLAN ID & Priority
Offset: 0x07 or decimal 7 – Port

Bits	Field	Type	Description
15:13	DefPri	RWR	Default Frame Priority. The bits of this register are used as the default Ingress priority to use when no other priority information is available (either the frame is not 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). These are the initial Frame priority (FPRI) bits assigned to the frame. The Ingress priority determines the Egress Queue the frame is switched into (the initial QPRI) by the DefPri bits being re-mapped by the IEEE-PRI Register (Port offset 0x18).
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 frame's VID are ignored and the DefaultVID below is assigned and replaced into the frame (if the frame egresses tagged – Egress Mode, Port offset 0x04). 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 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 internally to the switch only so 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 frames egress unmodified (Egress Mode, Port offset 0x04) or the internal VID will be added to the frame if it is set to egress tagged.

Table 15: Port Control 2 Register
Offset: 0x08 or decimal 8 – Port

Bits	Field	Type	Description
15	ForceGoodFCS	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 they will be 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 will be accepted by the switch (assuming the frame's length is good and it has a place to go).
14	AllowBad	RWR	Allow receiving frames on this port with a bad FCS. Allow receiving frames on this port with a bad FCS. When this bit is cleared to a zero the normal action of discarding ingressing frames with CRC errors occurs on this port. When this bit is set to a one frames received on this port with a CRC error are not discarded. Their CRC is not fixed, nor made good, either. Frames that are modified during ingress that are received with a CRC error will have a new 'bad' CRC placed into the frame. The frame's size must be good or the frame will still be discarded. All other switch actions remain the same including the Statistics counters which will still reflect the correct number of CRC errors received.
13:12	Jumbo Mode On the internal CPU's Port – a value of 0x0 or 0x1 is expected but not required (although larger frame size settings will not work as expected)	RWS to 0x2	Jumbo Mode. These bits determine the maximum frame size (or MTU) allowed to be received or transmitted from or to a given physical port. 0x0 = Receive and Transmit frames with max byte count of 1522 0x1 = Receive and Transmit frames with max byte count of 2048 0x2 = Receive and Transmit frames with max byte count of 10240 0x3 = Reserved This implies that a Jumbo frame may be allowed to be received from a given input port but may or may not be allowed to be transmitted out of a port or ports. For example if Port 2's JumboMode is 0x1 and it receives a 2100 Bytes frame, then the ingress pipe discards the frame as an oversized frame and the InOversize MIB counter is updated. Another example is if a Jumbo frame, say 4500 Bytes long, is destined to go to ports 3 and 4 and Port 3's JumboMode is 0x2 and Port 4's JumboMode is 0x0, then the frame gets sent to Port 3 but not to Port 4. **NOTE:** The definition of frame size is counting the frame bytes from the MAC-DA to the Layer2 CRC of the frame.

Table 15: Port Control 2 Register (Continued)
Offset: 0x08 or decimal 8 – Port

Bits	Field	Type	Description
11:10	802.1Q Mode	RWR	IEEE 802.1Q Mode for this port. These bits determine if 802.1Q based VLANs are used along with port based VLANs (Port offst 0x06) for this Ingress port. It also determines the action to be taken if an 802.1Q VLAN Violation is detected. VLAN barriers (both port based and 802.1Q based) can be bypassed by VLANTunnel (Port offset 0x04), MGMT (section ???) and AVB_Tunnel (AVB Policy Port offset 0x0) frames. These bits work as follows: **0x0 = 802.1Q Disabled.** Use Port Based VLANs only. The VLANTable bits (Port offset 0x06) 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) which is used as the VID in the Provider Tag if the frame egresses a Provider port (see Frame Mode, Port offset 0x04). **0x1 = Fallback.** Enable 802.1Q for this Ingress port. Do not discard Ingress Membership violations[2] and use the VLANTable bits (i.e., port based VLANs – Port offset 0x06) if the frame's VID is not contained in the VTU (both errors are logged – Global 1 offset 0x05). **0x2 = Check.** Enable 802.1Q for this Ingress port. Do not discard Ingress Membership violations but discard the frame if its VID is not contained in the VTU (both errors are logged – Global 1 offset 0x05). **0x3 = 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 – Global 1 offset 0x05).
9	Discard Tagged	RWR	Discard Tagged Frames. When this bit is set to a one all non-MGMT frames that are processed as tagged will be 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, Port offset 0x04) and this bit is set to a one, frames that contain an Ether Type that matches the port's PortEType (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 15: Port Control 2 Register (Continued)
Offset: 0x08 or decimal 8 – Port

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 will be 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, Port offset 0x04) and this bit is set to a one, frames that don't contain an Ether Type that matches the port's PortEType (Port offset 0x0F) and frames that contain an Ether Type that matches the port's PortEType 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 will occur where a frame's DA address is used to direct the frame out the correct port(s). When this bit is cleared to a zero the frame will be sent out the port(s) defined by EgressFloods (Port offset 0x04) even if the DA is found in the address database. **NOTE:** If a multicast 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 1 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. ARP Mirror should not be set on DSA or Ether Type DSA ports (see Frame Mode, Port offset 0x04) or extra mirroring will result.
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 be sent to the EgressMonitorDest Port (Global 1 offset 0x1A). 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 15: Port Control 2 Register (Continued)
Offset: 0x08 or decimal 8 – Port

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 in this port will also be sent to the IngressMonitorDest Port (Global 1 offset 0x1A). The frame will be sent to the IngressMonitorDest Port even if it is discarded due to switching policy (like VLAN membership, etc.) but the frame will not be forwarded if its contains an error (like CRC, etc.) or is filtered by ingress rate limiting (Global 2 offsets 0x09 and 0x0A). 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	Use Def Qpri	RWR	Use Default Queue Priority. When this bit is cleared to a zero the initial Qpri (Queue Priority) assigned to a frame is determined by the port's InitialPri and TagIfBoth settings (Port offset 0x04). When this bit is set to a one the initial Qpri assigned to the frame is the port's DefQPri (bits 2:1 below).
2:1	DefQPri	RWR	Default Queue Priority. The value of these bits are used as the Port's default queue priority and may be assigned to frames entering this port based on the UseDefQPri bit (above).
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 VLANs are supported at the same time, in-chp and cross-chip).
2. VTU or 802.1Q Ingress Membership violations occur when the source port is not a member of the frame's VID

Table 16: Egress Rate Control
Offset: 0x09 or decimal 9 – Port

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:8	Frame Overhead	RWR	Egress Rage 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 added to the Egress Rate Control's calculated transmitted byte count (Port offset 0x0A) meaning Egress Rate Control must be enabled for this Frame Overhead adjustment to work[1]. The egress overhead adjustment can add the following number of byte times to each frame's transmitted byte count: 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/transmitted byte count[2] 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 where Egress Rate is set to 64Kbps or 3906 or 0xF42. For any rate between 1Mbps and 100 Mbps: EgressDec = desired rate / 1 Mbps where Egress Rate is set to 1 Mbps or 250 or 0xFA. For any rate between 100Mbps and 1 Gbps: EgressDec = desired rate / 10 Mbps where Egress Rate is set to 10 Mbps or 25 or 0x19.

1. To increase every frame's IPG by the selected Frame Overhead bytes, enable the Egress Rate limiter (Port offset 0x0A) to match the link speed of the port. This enables the limiter (and the Frame Overhead bytes) without really limiting the rate of the port, other than to add the Frame Overhead bytes to every frame's IPG.
2. The transmitted byte count will equal the IPG increase in bytes if the Egress Rate limiter is enabled and set equal to the link speed of the port.

Table 17: Egress Rate Control 2
Offset: 0x0A or decimal 10 – Port

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: 0x0 = Frame based 0x1 = Count all Layer 1 bytes 0x2 = Count all Layer 2 bytes 0x3 = 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. 0x0 = Use an weighted round robin queuing scheme (default it 8, 4, 2, 1) 0x1 = Use Strict for priority 3 and use weighted round robin for priorities 2, 1 and 0 (default is 4, 2, 1) 0x2 = Use Strict for priorities 3 and 2 and use weighted round robin for priorities 1 and 0 (default is 2, 1) 0x3 = Use a Strict priority scheme for all priorities

Table 17: Egress Rate Control 2 (Continued)
Offset: 0x0A or decimal 10 – Port

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 / (32ns * Desired Egress Rate bits/sec) For example: CountMode = 0x2; Desired Rate = 640Kbps; EgressDec = 640Kbps / 64Kbps = 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 64Kbps to 1Gbps in the following increments: Desired rate between 64Kbps and 1Mbps in increments of 64Kbps. Desired rate between 1Mbps to 100Mbps in increments of 1Mbps. Desired rate between 100Mbps to 1Gbps in increments of 10Mbps. 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 / (32ns * 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 18: Port Association Vector
Offset: 0x0B or decimal 11 – Port

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 (section Table 71, ATU Operation Register, on page 102) 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 for 802.1x MAC authentication is enabled on this port. In this mode, an ATU Miss Violation interrupt will occur when a new SA address is received in a frame on this port. Automatic 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 18: Port Association Vector (Continued)
Offset: 0x0B or decimal 11 – Port

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 6.3) 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 will be 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.
10:7	Reserved	RES	Reserved for future use.
6:0	PAV	RWS to all zeros except for this port's bit	Port Association Vector for ATU learning. The value in these bits are 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 11, Port Control Register, on page 29). The PAV bits can be used to set up port trunking (along with the VLANTable bits (Table 13, Port Based VLAN Map, on page 38). For the two or more ports that form a trunk, set all of their port's bits to a one in all the port's PAV registers. Then use the VLANTable (Port offset 0x06) to isolate the ports from each other, or use the TrunkMask table (Global 2 offset 0x07) to steer the traffic from the other ports down the desired trunk line of the set.

Table 19: Port ATU Control[1]
Offset: 0x0C or decimal 12 – Port

Bits	Field	Type	Description
15	Read LearnCnt	RWR	Port's Auto Learning Limit or port's current Auto Learning count. Read the current number of 'active' unicast MAC addresses associated with this port. When this bit is cleared to a zero the LearnLimit RWR bits are accessible below. When this bit is set to a one the LearnCnt RO bits are accessible below. **NOTE:** Changing this bit from a 1 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) unless bit 12 is set to a one. This 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. Alternately, bit 12 can be set to a one to prevent bits 7:0 from overwriting the LearnLimit register setting. Likewise, a single write to this register that results with this bit being 1 must ensure bits 7:0 do NOT modify the LearnLimit register.
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.
12	KeepOld LearnLimit	RWR	Keep Old Learn Limit. When this bit is set to a zero and ReadLearnCnt (bit 15) is a zero, bits 7:0 will be written to the LearnLimit register. When this bit is set to a one, bits 7:0 will *not* be written to the LearnLimit register. This bit being a one allows the ReadLearnCnt bit (bit 15) to toggled without modifying the LearnLimit's register value.
11:10	Reserved	RES	Reserved for future use.

Table 19: Port ATU Control[1] (Continued)
Offset: 0x0C or decimal 12 – Port

Bits	Field	Type	Description
9:0	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[3]. In this mode the reading and writing of this register goes to the LearnLimit register (writing to the LearnLimit register is disabled if the KeepOldLearnLimit bit is set to a one). 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 greater than what it was before, this new value takes effect immediately. If the new LearnLimit value is set to a value less than or equal to the current LearnCtr, LimitReached will be set and no new addresses can be learned on the port. The pre-existing learned addresses that are over the limit will remain valid until they age out. 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, it is best to 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.
3. This hardware Learn Limit feature requires Learn2All must = 1 (Global offset 0x0A).

Table 20: Override Register
Offset: 0x0D or decimal 13 – Port

Bits	Field	Type	Description
15	DAQPri Override	RWR	DA Queue Priority Override. When this bit is cleared to a zero normal frame priority processing occurs. When this bit is set to a one then DA ATU queue priority overrides can occur on this port. A DA ATU queue 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 the MACPri 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. A 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 a DA queue priority override was used on the original network ingress port. The DA ATU queue Priority Override has highest priority over the port's Default queue Priority, the frame's IEEE and/or IP queue priorities, the VTU queue Priority Override and the SA queue Priority Override. **NOTE:** If a frame's DA is contained in the ATU with a MGMT Entry State the frame's queue priority will be overridden regardless of the state of this bit.
14	DAFPri Override	RWR	DA Frame Priority Override. When this bit is cleared to a zero normal frame priority processing occurs. When this bit is set to a one then DA ATU frame priority overrides can occur on this port. A DA ATU frame 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 the MACPri 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 PCP priority in the frame will be this new MAC_FPri value. DA frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, Port offset 0x04). The DA ATU frame Priority Override has highest priority over the port's Default frame Priority, the frame's IEEE and/or IP frame priorities, the VTU frame Priority Override and the SA frame Priority Override. **NOTE:** If a frame's DA is contained in the ATU with a MGMT Entry State the frame's Fpri will be overridden regardless of the state of this bit.

Table 20: Override Register (Continued)
Offset: 0x0D or decimal 13 – Port

Bits	Field	Type	Description
13	SAQPri Override	RWR	SA Priority Queue Override. When this bit is cleared to a zero normal frame priority processing occurs. When this bit is set to a one then SA ATU queue priority overrides can occur on this port. An SA ATU queue 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 the MACPri 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. A 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 an SA queue priority override was used on the original network ingress port. The SA ATU Priority Override has higher priority than the port's Default queue Priority, the frame's IEEE and/or IP queue priorities and the VTU queue Priority Override. The queue priority determined by the frame's SA can be overridden, however, by the frame's DA queue Priority Override.
12	SAFPri Override	RWR	SA Priority Frame Override. When this bit is cleared to a zero normal frame priority processing occurs. When this bit is set to a one then SA ATU frame priority overrides can occur on this port. An SA ATU frame 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 the MACPri 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 PCP priority in the frame will be this new MAC_FPri value. SA frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, Port offset 0x04). The SA ATU frame Priority Override has higher priority than the port's Default frame Priority, the frame's IEEE and/or IP frame priorities, and the VTU frame Priority Override. The priority determined by the frame's SA can be overridden, however, by the frame's DA frame Priority Override.

Table 20: Override Register (Continued)
Offset: 0x0D or decimal 13 – Port

Bits	Field	Type	Description
11	VTUQPri Override	RWR	VTU Queue Priority Override. When this bit is cleared to a zero normal frame priority processing occurs. When this bit is set to a one then VTU queue priority overrides can occur on this port. A VTU queue priority override occurs when the determined VID of a frame[1] results in a VID whose UseVIDQPri bit in the VLAN database is set to a one. When this occurs 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. A 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 a VID queue priority override was used on the original network ingress port. The VTU queue Priority Override has higher priority than the port's Default queue Priority and the frame's IEEE and/or IP queue priorities. The queue priority determined by the frame's VID can be overridden, however, by the frame's SA and/or DA queue Priority Overrides.
11:10	VTUFPri Override	RWR	VTU Frame Priority Override. When this bit is cleared to a zero normal frame priority processing occurs. When this bit is set to a one then VTU frame priority overrides can occur on this port. A VTU frame priority override occurs when the determined VID of a frame[2] results in a VID whose UseVIDFPri bit in the VLAN database is set to a one. When this occurs 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 PCP priority in the frame will be this new VID_FPri value. VID frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, Port offset 0x04). The VTU frame Priority Override has higher priority than the port's Default frame Priority and the frame's IEEE and/or IP frame priorities. The frame priority determined by the frame's VID can be overridden, however, by the frame's SA and/or DA frame Priority Overrides.
9	Mirror SA Miss	RWR	Mirror Source Address Misses to the MirrorDest port (Global 1 offset 0x1A). When this bits is cleared to a zero normal operation occurs. When this bit is set to a one and a Source Address search miss occurs (i.e., the frame's SA is not currently in the ATU), the frame is mirrored to the port pointed to by the Mirror Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including IRL, the port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolMirror (Policy Mirror) frame.

Table 20: Override Register (Continued)
Offset: 0x0D or decimal 13 – Port

Bits	Field	Type	Description
8	Mirror VTU Miss	RWR	Mirror VLAN Identifier Misses to the MirrorDest port (Global 1 offset 0x1A). When this bits is cleared to a zero normal operation occurs. When this bit is set to a one and a VID search miss occurs (i.e., the VID assigned to the frame is not currently in the VTU), the frame is mirrored to the port pointed to by the Mirror Dest register only (Global 1, offset 0x1A. If the frame is filtered by some other ingress policy (including IRL, the port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolMirror (Policy Mirror) frame.
7	Trap DA Miss	RWR	Trap Destination Address Misses to CPU. When this bits is cleared to a zero normal operation occurs. When this bit is set to a one and a Destination Address search miss occurs (i.e., the frame's DA is not currently in the ATU), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including IRL, the port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
6	Trap SA Miss	RWR	Trap Source Address Misses to CPU. When this bits is cleared to a zero normal operation occurs. When this bit is set to a one and a Source Address search miss occurs (i.e., the frame's SA is not currently in the ATU), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including IRL, the port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
5	Trap VTU Miss	RWR	Trap VLAN Identifier Misses to CPU. When this bits is cleared to a zero normal operation occurs. When this bit is set to a one and a VID search miss occurs (i.e., the VID assigned to the frame is not currently in the VTU), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including IRL, the port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
4	Trap TCAM Miss Valid for 88E6341 only	RWR	Trap TCAM Misses to CPU. When this bits is cleared to a zero normal operation occurs. When this bit is set to a one and a TCAM search miss occurs (i.e., no TCAM entry matched the frame), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including IRL, the port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
3:2	Reserved	RES	Reserved for future use.

Table 20: Override Register (Continued)
Offset: 0x0D or decimal 13 – Port

Bits	Field	Type	Description
1:0	TCAM Mode Only valid for devices that support a TCAM	RWR	TCAM Mode. These bits determine the TCAM's mode as follows: 0x0 = TCAM disabled (default) 0x1 = TCAM enabled for 48 byte searches only 0x2 = TCAM enabled for 48 byte and/or 96 bytes searches 0x3 = Reserved for future use **NOTE:** Do not change the value of these bits while frames are flowing into this port. You must first put the port in the Disable Port State (Port offset 0x04), then change these TCAM Mode bits, then re-enable the port's Port State bits. **NOTE:** For proper TCAM operation the TCAM Mode must match the size of the TCAM entries that will be used on this port. Specifically, if a TCAM entry is larger than the TCAM Mode setting on this port, that TCAM entry's VID and Load Balance Overrides cannot be used. **NOTE:** TCAM compares are performed in 48 byte increments. Therefore 64 to 95 byte frame sizes can match to 48 byte TCAM entries only.

1. The VID of a frame could be a tagged frame's VID or the port's DefaultVID.
2. The VID of a frame could be a tagged frame's VID or the port's DefaultVID.

Table 21: Policy Control Register[1]
Offset: 0x0E or decimal 14 – Port

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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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. **NOTE:** Frames with a Multicast SA cannot have SA Policy performed on them as Multicast SA's are not processed by the ATU. A TCAM entry can be used instead.

Table 21: Policy Control Register[1]
Offset: 0x0E or decimal 14 – Port

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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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 21: Policy Control Register[1]
Offset: 0x0E or decimal 14 – Port

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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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 21: Policy Control Register[1]
Offset: 0x0E or decimal 14 – Port

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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (Global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (Global offset 0x1A) 0x3 = 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 22: Port E Type
Offset: 0x0F or decimal 15 – Port

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 23: LED Control
Offset: 0x16 or decimal 22 – Port

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 10:0 will be loaded into the LED Control register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:12	Pointer	RWR	Pointer to the desired LED Control register. These bits select one of the possible LED Control registers for both read and write operations (but not all entries exist). A write operation occurs if the Update bit is a one. Otherwise a read operation occurs. Each valid Pointer value is described below: Port 0: 0x0 = Global LED Control 0x1 to 0x4 = Special LED 1 to 4 Configuration 0x5 to 0x6 = Reserved for future use 0x7 = Global LED Control 2 Ports 1 to 4: 0x0 = Control for LED 0 & 1 0x1 = Control for LED 2 & 3 0x2 to 0x5 = Reserved for future use 0x6 = Stretch and Blink Rate 0x7 = Reserved for future use
11	Reserved	RES	Reserved for future use
10:0	Data	RWR	LED Control data read or written to the register pointed to by the Pointer bits above.

The individual registers accessed by the LED Control register are described below and are indicated with a light yellow background heading color.

The LED control in the registers are organized as 4 LEDs per port for 4 ports. The physical LEDs on the device pins are organized as 4 Rows with 4 Columns. The Table below shows the port to physical LED mapping.

Table 24: Port to Physical Mapping

	C0_LED	C1_LED	C2_LED	C3_LED
R0_LED	Port 1 LED 0	Port 1 LED 1	Port 1 LED 2	Port 1 LED 3
R1_LED	Port 2 LED 0	Port 2 LED 1	Port 2 LED 2	Port 2 LED 3
R2_LED	Port 3 LED 0	Port 3 LED 1	Port 3 LED 2	Port 3 LED 3
R3_LED	Port 4 LED 0	Port 4 LED 1	Port 4 LED 2	Port 4 LED 3

Note: The device supports 16 LEDs or 4 LEDs per PHY port. Ports 0 & 5 do not have LED Control registers, but LEDs that are activated by these ports are supported via control registers on Ports 1 to 4.

The LED Link function is a Port's MAC Link which will track the Port's PHY Link (if a PHY is attached to the MAC internally or if the PPU is connected to an external PHY for a port via the MDC_PHY & MDIO_PHY pins). The only time the Port's MAC Link will not track the Port's PHY Link is when the Port's Link is being forced (Port offset 0x01).

Table 25: Global LED Control, Index: 0x00 of LED Control Ports 0 – Port Offset 0x16

Bits	Field	Type	Description
10:5	Reserved	RES	Reserved for future use.
4	Direct LEDs	RWR	Direct LED Mode Setting this bit to a one is used to select the LED direct drive mode which is a special case where Ports 1 to 4 LED 0 & Ports 1 to 3 LED 1 (7 LEDs total) are directly driven on different LED/EEPROM pins without any LED multiplexing (see the pin list for which pins are assigned to which ports). In this mode the pins are active low to light a LED. Although the LEDs are not multiplexed in this mode, the EEPROM can still be accessed.
3:0	Skip Columns	RWR or RWS to 0xF if the IMP boots	Skip LED Columns. The LED Columns can be selectively active or not. Setting the Column's bits to a one deactivates the LED Column. Clearing the Column's bit to a zero activates the LED Column. Bit 0 controls LED Column 0, bit 1 controls LED Column 1, bit 2 controls LED Column 2 and bit 3 controls LED Column 3. The brightness of the LEDs will increase as fewer LED Columns are active. This register is cleared to zero to light all LEDs during RESETn active. At the rising edge of RESETn this register is set to all ones allowing fast reading of the EEPROM. If there is no EEPROM or the 1st byte of the EEPROM is 0x76 (the Register Loader primer) and the EEPROM is done, then this register is cleared back to zero. In all other cases the IMP is in control and needs to determine when to turn on the LEDs by clearing these bits to zero. The IMP is given control over the LEDs as the speed of booting from the EEPROM is greatly reduced when any of the LED Columns are enabled.

Table 26: Special LED 1 Control, Index: 0x01 of LED Control on Port 0 – Port Offset 0x16

Bits	Field	Type	Description
10:0	LAN Link/Act	RWS to 0x01E	LAN Link Activity LED. Special LED 1 is a single Link Activity LED that is a combination of any of the selected port's Link Activity LED. If any of the selected port's Link is active this LED is on. If any of the selected Link'ed port's Activity is active this LED will blink off. This LED will be off if all the selected port's Links are down. This can be used as a front panel LED to indicate Link/Activity for any of the LAN ports. The default value selects Link/Activity for Ports 1 to 4 only. The bits of this register are used to define which ports are to be considered LAN ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a LAN port for purposes of this LED only. **NOTE:** This device supports Ports 0 to 6 only.

Table 27: Special LED 2 Control, Index: 0x02 of LED Control on Port 0 – Port Offset 0x16

Bits	Field	Type	Description
10:0	WAN Link/Act	RWS to 0x020	WAN Link Activity LED. Special LED 2 is a single Link Activity LED that is a combination of any of the selected port's Link Activity LED. If any of the selected port's Link is active this LED is on. If any of the selected Link'ed port's Activity is active this LED will blink off. This LED will be off if all the selected port's Links are down. This can be used as a front panel LED to indicate Link/Activity for any of the WAN ports. The default value selects Link/Activity for Port 5 only. The bits of this register are used to define which ports are to be considered WAN ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a LAN port for purposes of this LED only. **NOTE:** This device supports Ports 0 to 6 only.

Table 28: Special LED 3 Control, Index: 0x03 of LED Control on Port 0 – Port Offset 0x16

Bits	Field	Type	Description
10:0	CPU Link/Act	RWS to 0x001	CPU Link Activity LED. Special LED 3 is a single Link Activity LED that is a combination of any of the selected port's Link Activity LED. If any of the selected port's Link is active this LED is on. If any of the selected Link'ed port's Activity is active this LED will blink off. This LED will be off if all the selected port's Links are down. This can be used as a front panel LED to indicate Link/Activity for the CPU port. The default value selects Link/Activity for Port 0 only. The bits of this register are used to define which ports are to be considered CPU ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a LAN port for purposes of this LED only. **NOTE:** This device supports Ports 0 to 6 only.

Table 29: Special LED 4 Control, Index: 0x04 of LED Control on Port 0 – Port Offset 0x16

Bits	Field	Type	Description
10:0	PTP Activity	RWS to 0x040	PTP Activity LED. Special LED 4 is a single Activity LED that is a combination of any of the selected port's PTP Activity LED. If any of the selected port's PTP Activity is active this LED will blink. This LED will be off if there is no PTP Activity on the selected ports. This can be used as a front panel LED to indicate PTP Activity for any of the ports. The bits of this register are used to define which ports are to be considered PTP ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a PTP port for purposes of this LED only. **NOTE:** This device supports Ports 0 to 6 only.

Table 30: Global LED Control 2, Index: 0x07 of LED Control Ports 0 – Port Offset 0x16

Bits	Field	Type	Description
10:4	Reserved	RES	Reserved for future use.
3:0	BlinkTImes	RWS to 0x5	LED Link Up Blink TImes When Link/Act/Speed by Blink Rate[1] (off = no link, on = link, blink = activity, blink speed = link speed) is selected for and LED, that LED will blink 'n' times at each new Link Up, even if there is no activity on the wire. The number of times the LED blinks (i.e., the value of 'n') is controlled by this register.

1. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.

Table 31: LED 0 & 1 Control, Index: 0x00 of LED Control for Ports 1 to 4 – Port Offset 0x16

Bits	Field	Type	Description
10:8	Reserved	RES	Reserved for future use.
7:4	LED1 Select	RWR or RWS based on LED Config[1]	LED 1 Selection. These bits select LED[1]'s output as follows: 0x0 = Special LED 1 0x1 = 10/100 Link/Act (off = no link, on = 10 or 100 link, blink = activity) 0x2 = 10/100 Link (off = no link, on = 10 or 100 link) 0x3 = Gig Link (off = no link, on = Gig link) 0x4 = Special LED 2 0x5 = Link/Act (off = no link, on = link, blink = activity) 0x6 = 10/Gig Link/Act (off = no link, on = 10 or Gig link, blink = activity) 0x7 = 10/Gig Link (off = no link, on = 10 or Gig link) 0x8 = Gig Link/100 Blink (off = no link or 10 Link, blink=100 Link, on=Gig Link) 0x9 = 100 Link (off = no link, on = 100 link) 0xA = 100 Link/Act (off = no link, on = 100 link, blink = activity) 0xB = Duplex/Collision (off = half duplex, on = full duplex, blink = col) 0xC = PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On
3:0	LED0 Select	RWR or RWS based on LED Config[2]	LED 0 Selection. These bits select LED[0]'s output as follows: 0x0 = Link/Act/Speed by Blink Rate[3] (off = no link, on = link, blink = activity, blink speed = link speed[4]) 0x1 = 100/Gig Link/Act (off = no link, on = 100 or Gig link, blink = activity) 0x2 = Gig Link/Act (off = no link, on = Gig link, blink = activity) 0x3 = Link/Act (off = no link, on = link, blink = activity) 0x4 = Special LED 3 0x5 = Special LED 4 0x6 = Duplex/Collision (off = half duplex, on = full duplex, blink = col) 0x7 = 10/Gig Link/Act (off = no link, on = 10 or Gig link, blink = activity) 0x8 = Link (off = no link, on = link) 0x9 = 10 Link (off = no link, on = 10 link) 0xA = 10 Link/Act (off = no link, on = 10 link, blink = activity) 0xB = 100/Gig Link (off = no link, on = 100 or Gig link) 0xC = PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On

1. This register resets to a value of 0x2 to 0x3 based on the value of the LED_SEL configuration pin (see the Pinlist)
2. This register resets to a value of 0x2 to 0x3 based on the value of the LED_SEL configuration pin (see the Pinlist)
3. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
4. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 'n' times on each new link up, even if there is no activity, so the speed of the link can be observed.

Table 32: LED 2 & 3 Control, Index: 0x01 of LED Control for Ports 1 to 4 – Port Offset 0x16

Bits	Field	Type	Description
10:8	Reserved	RES	Reserved for future use.
7:4	LED3 Select	RWR or RWS based on LED Config[1]	LED 3 Selection. These bits select LED[3]'s output as follows: 0x0 = Special LED 1 0x1 = 10/100 Link/Act (off = no link, on = 10 or 100 link, blink = activity) 0x2 = Special LED x (where x = the port number) 0x3 = Special LED x (where x = the port number) 0x4 = Special LED 2 0x5 = Link/Act (off = no link, on = link, blink = activity) 0x6 = 10/Gig Link/Act (off = no link, on = 10 or Gig link, blink = activity) 0x7 = 10/Gig Link (off = no link, on = 10 or Gig link) 0x8 = Gig Link/100 Blink (off=no link or 10 Link, blink=100 Link, on=Gig Link) 0x9 = 100 Link (off = no link, on = 100 link) 0xA = 100 Link/Act (off = no link, on = 100 link, blink = activity) 0xB = P5 2500 Link (off = no link, on = 2500 link) 0xC = PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On
3:0	LED2 Select	RWR or RWS based on LED Config[2]	LED 2 Selection. These bits select LED[2]'s output as follows: 0x0 = Link/Act/Speed by Blink Rate[3] (off = no link, on = link, blink = activity, blink speed = link speed[4]) 0x1 = 100/Gig Link/Act (off = no link, on = 100 or Gig link, blink = activity) 0x2 = Activity (off = no link or no activity, blink = activity) 0x3 = 10/100 Link (off = no link, on = 10 or 100 link) 0x4 = Special LED 3 0x5 = Special LED 4 0x6 = Duplex/Collision (off = half duplex, on = full duplex, blink = col) 0x7 = P5 2500 Link (off = no link, on = 2500 link) 0x8 = P5 Link (off = no link, on = P5 link) 0x9 = P0 Link (off = no link, on = P0 link) 0xA = 10 Link/Act (off = no link, on = 10 link, blink = activity) 0xB = 100/Gig Link (off = no link, on = 100 or Gig link) 0xC = PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On

1. This register resets to a value of 0x2 to 0x3 based on the value of the LED_SEL configuration pin (see the Pinlist)
2. This register resets to a value of 0x2 to 0x3 based on the value of the LED_SEL configuration pin (see the Pinlist)
3. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
4. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 'n' times on each new link up, even if there is no activity, so the speed of the link can be observed.

Table 33: Stretch & Blink Rate, Index: 0x06 of LED Control Ports 1 to 4 – Port Offset 0x16

Bits	Field	Type	Description
10:7	Reserved	RES	Reserved for future use.
6:4	Pulse Stretch	RWS to 0x1	Pulse Stretch Selection for all the LED on this port. These bits select the port's LED stretch duration as follows: 0x0 = no pulse stretching 0x1 = 21 mSec (Default) 0x2 = 42 mSec 0x3 = 84 mSec 0x4 = 168 mSec 0x5 to 0x7 = Reserved for future use
3	Reserved	RES	Reserved for future use.
2:0	Blink Rate	RWS to 0x2	Blink Rate Selection for all the LEDs on this port. These bits select the port's LED blink rate as follows: 0x0 = 21 mSec 0x1 = 42 mSec 0x2 = 84 mSec (Default) 0x3 = 168 mSec 0x4 = 336 mSec 0x5 = 672 mSec 0x6 to 0x7 = Reserved for future use **NOTE:** If LED0 Select = 0x0 then these bits define the Blink Rate if the port links with a speed of Gig. If the port links to a speed of 100 then the blink time will be twice the value of this register and if the port links to a speed of 10 then the blink time will be four times the value of this register.

Table 34: Port IEEE Priority Remapping Registers
Offset: 0x18 or decimal 24 – Port

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 will be 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 will be 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 will be 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 will be written to the frame's tag.

Table 35: Port IEEE Priority Remapping Registers
Offset: 0x19 or decimal 25 – Port

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 will be 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 will be 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 will be 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 will be written to the frame's tag.

Table 36: Queue Counter Registers
Offset: 0x1B or decimal 27 – Port

Bits	Field	Type	Description
15:12	Mode	RWS to 0x8	Mode. The setting of these bits determines the content of the data returned in the Data field bits below.
11	Self Inc	RWR	Self Increment Mode. When this bit is cleared to a zero, the Mode bits above will remain constant after each read from this register. When this bit is set to a one, the Mode bits above will increment by one after each read to this register. This allows quicker reading of all the queue data from this register as the Mode bits do not need to be written between each read.
10:9	Reserved	RES	Reserved for future use.
8:0	Data	RO	Data. The data returned in this field is controlled by the Mode bits above as follows: When Mode = 0x0 to 0x3 → Return Egress Queue Size Counter for this port's Qpri 0x0 to 0x3 respectively. 0x4 to 0x7 → Return Egress Queue Size Counter for this port's Qpri 0x0 to 0x3 respectively (a mirror of the above counters so that the self incrementing Mode can be used more effectively). 0x8 → Return the Egress Total Queue Size Counter for this port. This counter reflects the current number of Egress buffers switched to this port. This is the total number of buffers across all priority queues. 0x9 → Return the Ingress Reserved Queue Size Counter for this port. This counter reflects the current number of reserved Ingress buffers assigned to this port. 0xA → Return BufHigh in bit 1 and Fc_En in bit 0. BufHigh is an output from the QC telling the MAC that it should perform Flow Control. Fc_En is an input into the QC telling it that Flow Control is enabled on this port. 0xB to 0xF → Reserved for future use. Returns zeros.

Table 37: Port Queue Control Register
Offset: 0x1C or decimal 28 – Port

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 Port Queue Control register referenced by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	Pointer to the desired octet of Port Queue Control. These bits select one of the possible Port Queue Control registers for both read and write operations. A write operation occurs if the Update bit is a one (the registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the desired register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 = Reserved 0x01= Filter QPri Enables 0x02 to 0x03 = Per Port Scratch registers 0x04 to 0x0F = Reserved for future use 0x10 to 0x17 = Hard Queue Limit registers 0x18 to 0x4F = Reserved for future use 0x50 to 0x53 = Begin & End Flow Control Thresholds & Port Counter 0x54 to 0x7F = Reserved for future use
7:0	Data	RWR	Octet Data of the Port Queue Control register. As referenced by the Pointer bits above.

The individual registers accessed by the Port Queue Control register are described below and are indicated with a light purple background heading color.

Table 38: Filter Queue Priorities, Index: 0x01 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	FilterQpri3	RWR	Egress Filter Qpri frames from Queue 3. See description for bit 0 below.
2	FilterQpri2	RWR	Egress Filter Qpri frames from Queue 2. See description for bit 0 below.
1	FilterQpri1	RWR	Egress Filter Qpri frames from Queue 1. See description for bit 0 below.
0	FilterQpri0	RWR	Egress Filter Qpri frames from Queue 0. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one all frames that are mapped to this port's egress queue 0 will NOT be transmitted (when it is the frame's turn to be transmitted out the port, the frame's buffers will be returned instead as if the link was down on the port – so this filtering is done on the output side of the Egress queue).

Table 39: Port Scratch 0, Index: 0x02 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7:0	Port Scratch 0	RWR	Port Scratch bits. These bits are 100% available to software for whatever purpose desired. These bits do not connect to any hardware function. NOTE: These bits are cleared to zero with a hardware reset.

Table 40: Port Scratch 1, Index: 0x03 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7:0	Port Scratch 1	RWR	Port Scratch bits. These bits are 100% available to software for whatever purpose desired. These bits do not connect to any hardware function. **NOTE:** These bits are cleared to zero with a hardware reset.

Table 41: Hard Queue Limit 0, Index: 0x10 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7	HardOnly0	RWR	Hard queue Limit Mode for Queue 0 on this port. When this bit is cleared to a zero the normal Queue Controller's Dynamic Queue limits are used on this queue on this port. The Dynamic Queue limits may be used in concert with the Hard Queue limits, if Hard Queue limits are enabled on this queue (see HardLimit for this queue below). When this bit is set to a one the Queue Controller's Dynamic Queue limits will be ignored by the Queue Controller. When this bit is set to a one the HardLimit register for this queue, below, must be non-zero. This configuration configures the queue for Hard Queue limits only.
6:0	HardLimit0	RWR	Hard queue Limit for Queue 0 on this port. When these bits are cleared to zero, Hard Queue limits are disabled on this queue. This default setting, along with the default setting of the HardOnly bit for this queue above, enabled the normal Dynamic Queue limits as the only queue limiting method used on the queue. When these bits are non-zero, they become the hard queue limit. The size of this queue will not exceed the value set in this register in terms of buffer counts. This is handled by the Queue Controller by adding the current frame's buffer count to the queue's current buffer count. If that sum exceeds the value currently set in this register the frame will not be mapped to this queue. Hard queue limits can work together with the Dynamic Queue limits or by itself as determined by the HardOnly bit for this queue above.

Note: Three queue limiting modes are supported as follows:

- Dynamic queue limits only (when HardOnly = 0 and HardLimit = 0)
- Hard queue limits only (when HardOnly = 1 and HardLimit <> 0)
- Both Dynamic and Hard queue limits together (when HardOnly = 0 and HardLimit <> 0)

When both Dynamic and Hard queue limits are used together, they work in an 'OR' fashion, meaning and frame will NOT be mapped to a port if either the Dynamic OR the Hard queue limits indicate the frame should NOT be mapped.

Table 42: Hard Queue Limit 1, Index: 0x11 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7	HardOnly1	RWR	Hard queue Limit Mode for Queue 1 on this port. When this bit is cleared to a zero the normal Queue Controller's Dynamic Queue limits are used on this queue on this port. The Dynamic Queue limits may be used in concert with the Hard Queue limits, if Hard Queue limits are enabled on this queue (see HardLimit for this queue below). When this bit is set to a one the Queue Controller's Dynamic Queue limits will be ignored by the Queue Controller. When this bit is set to a one the HardLimit register for this queue, below, must be non-zero. This configuration configures the queue for Hard Queue limits only.
6:0	HardLimit1	RWR	Hard queue Limit for Queue 1 on this port. When these bits are cleared to zero, Hard Queue limits are disabled on this queue. This default setting, along with the default setting of the HardOnly bit for this queue above, enabled the normal Dynamic Queue limits as the only queue limiting method used on the queue. When these bits are non-zero, they become the hard queue limit. The size of this queue will not exceed the value set in this register in terms of buffer counts. This is handled by the Queue Controller by adding the current frame's buffer count to the queue's current buffer count. If that sum exceeds the value currently set in this register the frame will not be mapped to this queue. Hard queue limits can work together with the Dynamic Queue limits or by itself as determined by the HardOnly bit for this queue above.

Table 43: Hard Queue Limit 2, Index: 0x12 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7	HardOnly2	RWR	Hard queue Limit Mode for Queue 2 on this port. When this bit is cleared to a zero the normal Queue Controller's Dynamic Queue limits are used on this queue on this port. The Dynamic Queue limits may be used in concert with the Hard Queue limits, if Hard Queue limits are enabled on this queue (see HardLimit for this queue below). When this bit is set to a one the Queue Controller's Dynamic Queue limits will be ignored by the Queue Controller. When this bit is set to a one the HardLimit register for this queue, below, must be non-zero. This configuration configures the queue for Hard Queue limits only.
6:0	HardLimit2	RWR	Hard queue Limit for Queue 2 on this port. When these bits are cleared to zero, Hard Queue limits are disabled on this queue. This default setting, along with the default setting of the HardOnly bit for this queue above, enabled the normal Dynamic Queue limits as the only queue limiting method used on the queue. When these bits are non-zero, they become the hard queue limit. The size of this queue will not exceed the value set in this register in terms of buffer counts. This is handled by the Queue Controller by adding the current frame's buffer count to the queue's current buffer count. If that sum exceeds the value currently set in this register the frame will not be mapped to this queue. Hard queue limits can work together with the Dynamic Queue limits or by itself as determined by the HardOnly bit for this queue above.

Table 44: Hard Queue Limit 3, Index: 0x13 of Queue Control – Port Offset 0x1C

Bits	Field	Type	Description
7	HardOnly3	RWR	Hard queue Limit Mode for Queue 3 on this port. When this bit is cleared to a zero the normal Queue Controller's Dynamic Queue limits are used on this queue on this port. The Dynamic Queue limits may be used in concert with the Hard Queue limits, if Hard Queue limits are enabled on this queue (see HardLimit for this queue below). When this bit is set to a one the Queue Controller's Dynamic Queue limits will be ignored by the Queue Controller. When this bit is set to a one the HardLimit register for this queue, below, must be non-zero. This configuration configures the queue for Hard Queue limits only.
6:0	HardLimit3	RWR	Hard queue Limit for Queue 3 on this port. When these bits are cleared to zero, Hard Queue limits are disabled on this queue. This default setting, along with the default setting of the HardOnly bit for this queue above, enabled the normal Dynamic Queue limits as the only queue limiting method used on the queue. When these bits are non-zero, they become the hard queue limit. The size of this queue will not exceed the value set in this register in terms of buffer counts. This is handled by the Queue Controller by adding the current frame's buffer count to the queue's current buffer count. If that sum exceeds the value currently set in this register the frame will not be mapped to this queue. Hard queue limits can work together with the Dynamic Queue limits or by itself as determined by the HardOnly bit for this queue above.

Table 45: Input Port Counter, Index: 0x53 of Flow Control – Port Offset 0x1C

Bits	Field	Type	Description
7:0	RxCount	RO	Receive Counter for this port When the port is configured for Port based Flow Control (the default setting when flow control is enabled – Port offset 0x02) this is the port's input queue count for all priorities used for determining when to assert and de-assert flow control (see Index 0x50 above). In this mode this register returns bits [8:1] of the Port's total counter, therefore the thresholds (Index 0x50 above) are in even numbers as well (in this mode). **NOTE:** This counter always reflects the input Queue count for this port (even if flow control is disabled).

Table 46: Port Queue Control 2 Register
Offset: 0x1D or decimal 29 – Port

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 Port Queue Control register referenced by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	Pointer to the desired octet of Port Queue Control 2. These bits select one of the possible Port Queue Control 2 registers for both read and write operations. A write operation occurs if the Update bit is a one (the registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the desired register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 & 0x01 = MaxSizeQ0 0x02 & 0x03 = MaxSizeQ1 0x04 & 0x05 = MaxSizeQ2 0x06 & 0x07 = MaxSizeQ3 0x08 to 0x7F = Reserved for future use
7:0	Data	RWR	Octet Data of the Port Queue Control 2 register. As referenced by the Pointer bits above.

The individual registers accessed by the Port Queue Control 2 register are described below and are indicated with a orange background heading color.

Table 47: Max Size Queue 0 Low, Index: 0x00 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7:0	MaxSizeQ0 Low Byte	RWR	Maximum Frame Size Queue 0 – Low Byte. These bits represent the lower 8 bits of a 15 bit Maximum Frame Size register. When this register is cleared to zero, maximum frame size checking for this queue is disabled and all frames are transmitted from this queue that do not exceed the limit for this port (by Jumbo Mode in Port offset 0x08). When this register is non-zero, it defines the maximum frame size that can be transmitted from this queue. Any frame that reaches this queue's head-of-line whose size is larger than this register setting will be discarded instead of being transmitted. Discarded frames are counted in the port's OutDropped Stat Counter (Global 1 offset 0x1D).

Table 48: Max Size Queue 0 High, Index: 0x01 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	MaxSizeQ0 Low Byte	RWR	Maximum Frame Size Queue 0 – High Byte. These bits represent the upper 7 bits of a 15 bit Maximum Frame Size register. See the full description of this register above.

Table 49: Max Size Queue 1 Low, Index: 0x02 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7:0	MaxSizeQ1 Low Byte	RWR	Maximum Frame Size Queue 1 – Low Byte. These bits represent the lower 8 bits of a 15 bit Maximum Frame Size register. When this register is cleared to zero, maximum frame size checking for this queue is disabled and all frames are transmitted from this queue that do not exceed the limit for this port (by Jumbo Mode in Port offset 0x08). When this register is non-zero, it defines the maximum frame size that can be transmitted from this queue. Any frame that reaches this queue's head-of-line whose size is larger than this register setting will be discarded instead of being transmitted. Discarded frames are counted in the port's OutDropped Stat Counter (Global 1 offset 0x1D).

Table 50: Max Size Queue 1 High, Index: 0x03 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	MaxSizeQ1 Low Byte	RWR	Maximum Frame Size Queue 1 – High Byte. These bits represent the upper 7 bits of a 15 bit Maximum Frame Size register. See the full description of this register above.

Table 51: Max Size Queue 2 Low, Index: 0x04 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7:0	MaxSizeQ2 Low Byte	RWR	Maximum Frame Size Queue 2 – Low Byte. These bits represent the lower 8 bits of a 15 bit Maximum Frame Size register. When this register is cleared to zero, maximum frame size checking for this queue is disabled and all frames are transmitted from this queue that do not exceed the limit for this port (by Jumbo Mode in Port offset 0x08). When this register is non-zero, it defines the maximum frame size that can be transmitted from this queue. Any frame that reaches this queue's head-of-line whose size is larger than this register setting will be discarded instead of being transmitted. Discarded frames are counted in the port's OutDropped Stat Counter (Global 1 offset 0x1D).

Table 52: Max Size Queue 2 High, Index: 0x05 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	MaxSizeQ2 Low Byte	RWR	Maximum Frame Size Queue 2 – High Byte. These bits represent the upper 7 bits of a 15 bit Maximum Frame Size register. See the full description of this register above.

Table 53: Max Size Queue 3 Low, Index: 0x06 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7:0	MaxSizeQ3 Low Byte	RWR	Maximum Frame Size Queue 3 – Low Byte. These bits represent the lower 8 bits of a 15 bit Maximum Frame Size register. When this register is cleared to zero, maximum frame size checking for this queue is disabled and all frames are transmitted from this queue that do not exceed the limit for this port (by Jumbo Mode in Port offset 0x08). When this register is non-zero, it defines the maximum frame size that can be transmitted from this queue. Any frame that reaches this queue's head-of-line whose size is larger than this register setting will be discarded instead of being transmitted. Discarded frames are counted in the port's OutDropped Stat Counter (Global 1 offset 0x1D).

Table 54: Max Size Queue 3 High, Index: 0x07 of Queue Control 2 – Port Offset 0x1D

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	MaxSizeQ3 Low Byte	RWR	Maximum Frame Size Queue 3 – High Byte. These bits represent the upper 7 bits of a 15 bit Maximum Frame Size register. See the full description of this register above.

Table 55: Cut Through Register
Offset: 0x1E or decimal 30 – Port

Bits	Field	Type	Description
15:12	Enable Select (valid on Port 0 only)	RWS	Port Enable Select. These bits are used to select the Px_ENABLE from one of the GPIO pins. It is the responsibility of the user to ensure that the selected GPIO pin is enabled to be a GPIO pin (see GPIO Configuration register, Global 2 Scratch & Misc Index 0x60 & 0x61), and that the GPIO pin is configured to be an input (see GPIO Direction register, Global 2 Scratch & Misc Index 0x62 & 0x63). Enable Select works as follows: 0x0 to 0xA = This port's Px_ENABLE connects to GPIO[Enable Select][1] 0xB to 0xE = Reserved for future use. 0xF = This port's Px_ENABLE connects to a one (enabled)[2]
11:9	Reserved	RES	Reserved for future use.
8	Cut Through This register has no effect on the internal CPU's Port – as that port does not do Cut Through.	RWR	Cut Through enable. When this bit is cleared to a zero, cut through from this Ingress port cannot occur. When this bit is set to a one, frames are allowed to cut through from this Ingress port to any port and queue whose Cut Through Queue bit (below) is set to a one. Many other conditions are required before a frame can actually be cut through the switch fabric, bypassing the normal Store and Forward Queue Controller. Some (but not all) of these conditions include: Both ports must be in the Forwarding Port State (Port offset 0x04), Ingress and Egress ports must be at the same speeds and both must be in full duplex mode (Port offset 0x00). The Egress MAC must not be transmitting and the target priority queue must be empty. **NOTE:** If Cut Through is enabled on a port that port should not enable IPv6 Snooping (Port offset 0x04) as the IPv6 Snoop point may be after byte 64. **NOTE:** When this bit is set to a one Energy Efficient Ethernet (EEE) on this port will be automatically prevented from entering the Low Power Idle state, effectively disabling EEE on this port.
7:4	Reserved	RES	Reserved for future use.
3:0	Cut Through Queue This register has no effect on the internal CPU's Port – as that port does not do Cut Through.	RWR	Cut Through Queues. When all of these bits are cleared to zero, cut through to this Egress port cannot occur. When any one of these bits are set to a one, frames are allowed to cut through to this Egress port, but only if the frame came from a Cut Through enabled Ingress port (see Cut Through bit above) and the frame is mapped to a priority queue whose Cut Through Queue register bit is set to a one. Bit 0 of this registers is the enable for Pri 0, bit 1 for Pri 1, etc.

1. An EnableSelect value of 0x0 selects GPIO[0], a value of 0x1 selects GPIO[1], a value of 0xA selects GPIO[10], etc.
2. The default value of 0xF sets the port's Px_ENABLE to one enabling the link on the port and drives the port's output pins. But if the port's Px_MODE (see the port's Px_OUTD pin descriptions) or C_Mode (Port offset 0x00) = 0x6 the port will be disabled overriding the port's Px_ENABLE.

Table 56: Debug Counter
Offset: 0x1F or decimal 31 – Port

Bits	Field	Type	Description
15:8	RxBad Frames/ Tx Collisions	RO	Bad Counter. When CtrMode is cleared to a zero (Global 1 offset 0x1C) this counter increments each time a frame enters this port that was an error on the wire. It does not matter if the frame's CRC is fixed by ForceGoodFCS (Port offset 0x08) being set to a one, this counter will still increment. A CRC error frame is one that is 64 bytes to MaxFrameSize (Global 1, offset 0x04) with a bad CRC (including alignment errors but not dribbles). Fragments and properly formed frames are not counted. When CtrMode is set to a one this counter increments each time a transmit collision occurs on this port. The counter will wrap around back to zero. The only time this counter will not increment is when this port is Disabled (see PortState in Port offset 0x04). This register can be cleared by changing the state of the CtrMode bit in Global Control 2 (Global 1 offset 0x1C) or by a Flush All Counters for this port or all ports StatsOp command (Global 1 offset 0x1D).
7:0	RxGood Frames/ Tx Transmit Frames	RO	Good Counter. When CtrMode is cleared to a zero (Global 1 offset 0x1C) this counter increments each time a frame enters this port that was not an error frame on the wire. It does not matter if the frame was filtered or discarded, only that the frame was received as good on the wire (i.e., its wire size is in the range of 64 bytes to MaxFrameSize (Global 1, offset 0x04) and its CRC was good. When CtrMode is set to a one this counter increments each time a frame is transmitted out this port. The counter will wrap around back to zero. The only time this counter will not increment is when this port is Disabled (see PortState in Port offset 0x04). This register can be cleared by changing the state of the CtrMode bit in Global Control 2 (Global 1 offset 0x1C) or by a Flush All Counters for this port or all ports StatsOp command (Global 1 offset 0x1D).

1.3.2 Switch Global 1 Registers

The device contains global registers that effect all the Ethernet ports in the device. Each global register is 16-bits wide and their bit assignments are shown in Table 57.

Table 57: Global 1 Register bit Map (Device Addr 0x1B)

Global Register Data Bits

SMI Register Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	IS	Reserved			IR	RES	Interrupt Status										0	Global Status ← IS = Init State, IR = Init Ready
1	Reserved								FID[7:0] (for ATU)								1	ATU FID
2	Reserved			VP	Reserved				FID[7:0] (for VTU)								2	VTU FID ← VP = Vid Policy
3	DL	VTU Unknown Filters			Reserved				SID[5:0] (for VTU)								3	VTU SID ← DL = Don't Learn
4	SR	R	DE	AB	Reserved		Interrupt Masks										4	Global Control ← SR = Software Reset, DE = Discard Excessive, AB = ARPwoBC
5	VB	VTUOp			Reserved				Violations	R	SPID						5	VTU
6	Reserved			V	VID												6	VTU ← V = Valid
7	P3's VTU/STU Data				P2s VTU/STU Data				P1's VTU/STU Data				P0's VTU/STU Data				7	VTU
8	Reserved				P6's VTU/STU Data				P5's VTU/STU Data				P4s VTU/STU Data				8	VTU
9	VP	VIDPri			Reserved												9	VTU
A	Reserved				Age TIme								L2	LA	Hash Select		10	ATU ← L2 = Learn2All, LA = LearnAvb
B	AB	ATUOp				MACPri			Violations				Reserved				11	ATU ← AB = ATU Busy
C	L	Reserved				Port Vector							EntryState/SPID				12	ATU ← L = LAG
D	ATU MAC Address Reg Bytes 0 & 1																13	ATU
E	ATU MAC Address Reg Bytes 2 & 3																14	ATU
F	ATU MAC Address Reg Bytes 4 & 5																15	ATU
10	Reserved																16	
11	Reserved																17	
12	Reserved																18	
13	Reserved																19	
14	Reserved																20	
15	Reserved																21	
16	Reserved																22	
17	Reserved																23	
18	IEEE PRI Mapping Reg																24	IEEE FPri to QPri Mapping
19	IP PRI Mapping Table																25	IP to QPri & FPri Mapping
1A	U		Pointer						Data								26	Monitor & MGMT Control ← U = Update
1B	Reserved								FreeQSize								27	Free Queue Size
1C	Header Type		RES		D C	RMU Mode			Hist		CM	Dev_Num					28	Global Control 2 ← DC = DA Check on Remote MGMT, CM = Counter Mode
1D	SB	StatsOp				BK		StatsPort				StatsPrt					29	Stats Operation ← SB = Stats Busy, BK = Stats Bank
1E	Stats Counter Register Byte 3 & 2																30	
1F	Stats Counter Register Byte 1 & 0																31	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		

Table 58: Switch Global Status Register
Offset: 0x00 or decimal 0 – G1

Bits	Field	Type	Description
15	InitState	RO	Initialization State. This bit indicates the initialization condition/state of the PHY Polling Unit (PPU) as follows: 0 = PPU is Active detecting and initializing external PHYs. The PortStatus registers (Port offset 0x00) must not be written to by software. 1 = PPU Polling. This indicates the PPU is Active polling the external PHYs. Software can write to the PortStatus registers (Port offset 0x00).
14:12	Reserved	RES	Reserved for future use.
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 are done with their initialization and are ready to accept frames.
10:9	Reserved	RES	Reserved for future use.
8	AVBInt Only valid for devices that support AVB/TSN	RO	AVB Interrupt. There are two entities that can set this bit. The PTP block interrupt (Global 2 offset 0x16 and PTP Global status register) or the Qav block interrupt (Global 2 offset 0x16 and Qav Global status registers). After reading the appropriate block registers, CPU clears the PTPInt bits which in turns clears this bit. This bit being high will cause the device's INTn pin to go low if the AVBIntEn bit (Global 1 offset 0x04) is set to a one.
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	ROC	Statistics Done Interrupt. This bit is set to a one whenever the STATBusy bit (Global 1 offset 0x1D) transitions from a one to a zero. It is automatically cleared when read. This bit being high will cause the device's INTn pin to go low if the STATDoneIntEn bit (Global 1 offset 0x04) 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 the pending VTU Violations have been serviced by the VTU Get/Clear Violation Data operation (Global 1 offset 0x05). This bit being high will cause the device's INTn pin to go low if the VTUProbIntEn bit (Global 1 offset 0x04) is set to a one.

Table 58: Switch Global Status Register (Continued)
Offset: 0x00 or decimal 0 – G1

Bits	Field	Type	Description
4	VTUDone	ROC	VTU Done Interrupt. This bit is set to a one whenever the VTUBusy bit (Global 1 offset 0x05) transitions from a one to a zero. It is automatically cleared when read. This bit being high will cause the device's INTn pin to go low if the VTUDoneIntEn bit (Global 1 offset 0x04) is 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 (Global 1 offset 0x0B). This bit being high will cause the device's INTn pin to go low if the ATUFullIntEn bit (Global 1 offset 0x04) is set to a one.
2	ATUDone	ROC	ATU Done Interrupt. This bit is set to a one whenever the ATUBusy bit (Global 1 offset 0x0B) transitions from a one to a zero. It is automatically cleared when read. This bit being high will cause the device's INTn pin to go low if the ATUDoneIntEn bit (Global 1 offset 0x04) is set to a one.
1	TCAM Int Only valid for devices that support a TCAM	ROC	TCAM Interrupt. This bit is set to a one whenever the TCAM gets a hit where the Action Int bit is set to a one (TCAM Page 2 offset 0x02). It is automatically cleared when read. This bit being high will cause the device's INTn pin to go low if the TCAMIntEn bit (Global 1 offset 0x04) is set to a one.
0	EEInt	ROC	EEPROM Done Interrupt. This bit is set to a one after the EEPROM is done loading registers or when an EEPROM operation is done (see EEPROM Control, Global 2 offset 0x14) and it is automatically cleared when read. This bit being high will cause the device's INTn pin to go low if the EEIntEn bit (Global 1 offset 0x04) is set to a one.

Table 59: ATU FID Register
Offset: 0x01 or decimal 1 – G1

Bits	Field	Type	Description
15:8	Reserved	RES	Reserved for future use.
7:0	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 60: VTU FID Register
Offset: 0x02 or decimal 2 – G1

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).
11:8	Reserved	RES	Reserved for future use.
7:0	FID	RWR	VTU MAC Address Forwarding Information Database (FID) number. On VTU Load and VTU GetNext <<#ount#ions>>, 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).

Table 61: VTU SID Register
Offset: 0x03 or decimal 3 – G1

Bits	Field	Type	Description
15	DontLearn	RWR	Don't Learn. When this bit is set to a one and this VTU entry is accessed by a frame, then learning will not take place on this frame. **NOTE:** This bit is present in the VTU only. On STU writes this bit is don't care and on STU reads there return zero.
14	FilterUC	RWR	Filter Unicast frames. When this bit is set to a one and this VTU entry is accessed by a frame, then that frame will be discarded (filtered) if the frame's DA is a Unicast address which is not found in the address database (ATU). **NOTE:** This bit is present in the VTU only. On STU writes this bit is don't care and on STU reads there return zero.
13	FilterBC	RWR	Filter Broadcast frames. When this bit is set to a one and this VTU entry is accessed by a frame, then that frame will be discarded (filtered) if the frame's DA is the Broadcast address which is not found in the address database (ATU). **NOTE:** This bit is present in the VTU only. On STU writes this bit is don't care and on STU reads there return zero.
12	FilterMC	RWR	Filter Mulicast frames. When this bit is set to a one and this VTU entry is accessed by a frame, then that frame will be discarded (filtered) if the frame's DA is a Multicast address (but not including the Broadcast address) which is not found in the address database (ATU). **NOTE:** This bit is present in the VTU only. On STU writes this bit is don't care and on STU reads there return zero.
11:6	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 62: Switch Global Control Register
Offset: 0x04 or decimal 4 – G1

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 will not be re-read. The ATU, VTU, MIBs, PHYs etc. are not effected by this bit. When the reset operation is done, this bit will be cleared to a zero automatically. The reset will occur immediately. So to prevent transmission of CRC frames, all the ports should be set to the Disabled state (Port offset 0x04), and wait for 2 mSec (i.e., the time for a maximum frame to be transmitted at 10 Mbits/sec) before the SWReset bit is set 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	ARPwo BC	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 (0x0806) and makes the decision. When disabled the switch core checks for both the Ether Type and Broadcast MAC address for ARP frame detection.
11:9	Reserved	RES	Reserved for future use.
8	AVBIntEn Only valid for devices that support AVB/TSN	RES	AVB Interrupt Enable. This bit must be set to a one to allow active interrupts enabled in AVB registers (Global 2 offset 0x16 & 0x17) to drive the device's INTn pin low.
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 VT Problem interrupt to drive the device's INTn pin low.

Table 62: Switch Global Control Register (Continued)
Offset: 0x04 or decimal 4 – G1

Bits	Field	Type	Description
4	VTUDone IntEn	RWR	VLAN Table Operation Done Interrupt Enable. This bit must be set to a one to allow the VT Done interrupt to drive the device's INTn pin low.
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	TCAM IntEn Only valid for devices that support a TCAM	RWR	TCAM Int Interrupt Enable. This bit must be set to a one to allow the TCAM Int interrupt 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 63: VTU Operation Register
Offset: 0x05 or decimal 5 – G1

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 (Global 1 offset 0x04).
14:12	VTUOp	RWR	VLAN Table Unit Table Opcode. The device supports the following VTU operations (all of these operations can be executed while frames are transiting through the switch): 0x0 = No Operation 0x1 = Flush All Entries in the VTU and the STU 0x2 = Reserved 0x3 = VTU Load[1] or Purge[2] an Entry 0x4 = VTU Get Next[3] 0x5 = STU Load[4] or Purge[5] an Entry 0x6 = STU Get Next[6] 0x7 = Get/Clear Violation Data[7]
11:7	Reserved	RES	Reserved for future use
6	Member Violation	RO	Source Port 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 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 1st 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 1st Miss Violation or Member Violation (above) will be saved until cleared.
4	Reserved	RES	Reserved for future use
3:0	SPID	RO	Source Port ID. 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 violation was the CPU's registers interface (i.e., the VTU was full during a CPU Load operation).

1. A VTU Entry is Loaded if 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 fields and it uses them as data to be loaded along with the desired VID and its port member data.
2. A VTU Entry is Purged if it exists and the Valid bit (in the VTU VID 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 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 a 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) the Get/Clear Violation VTUOp can be used to retreive 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. When all Violations currently pending in the VTU have been serviced the VTUProb bit in Global Status will be cleared to a zero.

Table 64: VTU VID Register
Offset: 0x06 or decimal 6 – G1

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 one's 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 commands and it is the VID that is associated with the VTU data below or the VID that caused the VTU Violation.

Table 65: VTU/STU Data Register Ports 0 to 3 for VTU Ops
Offset: 0x07 or decimal 7 – G1

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
13:12	Member TagP3	RWR	Ingress and Egress Membership and Egress Tagging for Port 3. These bits are used to support 802.1Q Egress 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	Ingress and Egress Membership and Egress Tagging for Port 2. These bits are used to support 802.1Q Egress 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	Ingress and Egress Membership and Egress Tagging for Port 1. These bits are used to support 802.1Q Egress 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	Ingress and Egress Membership and Egress Tagging for Port 0. These bits are used to support 802.1Q Ingress and Egress membership and Egress Tagging as follows: 0x0 = Port is a member of this VLAN & frames Egress unmodified. 0x1 = Port is a member of this VLAN & frames Egress Untagged. 0x2 = Port is a member of this VLAN & frames Egress Tagged. 0x3 = 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 66: VTU/STU Data Register Ports 0 to 3 for STU Ops
Offset: 0x07 or decimal 7 – G1

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	RWR	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	RWR	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: 0x0 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 0x1 = Blocking/Listening Port State for this port for frames with this VID. 0x2 = Learning Port State for this port for frames with this VID. 0x3 = 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 67: VTU/STU Data Register Ports 4 to 5 for VTU Ops
Offset: 0x08 or decimal 8 – G1

Bits	Field	Type	Description
15:10	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
9:8	Member TagP6	RWR	Ingress and Egress Membership and Egress Tagging for Port 6. These bits are used to support 802.1Q Egress 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	Ingress and Egress Membership and Egress Tagging for Port 5. These bits are used to support 802.1Q Egress 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: 0x0 = Port is a member of this VLAN and frames are to Egress unmodified. 0x1 = Port is a member of this VLAN and frames are to Egress Untagged. 0x2 = Port is a member of this VLAN and frames are to Egress Tagged. 0x3 = 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 68: VTU/STU Data Register Ports 4 to 5 for STU Ops
Offset: 0x08 or decimal 8 – G1

Bits	Field	Type	Description
15:12	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	RWR	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	RWR	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: 0x0 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 0x1 = Blocking/Listening Port State for this port for frames with this VID. 0x2 = Learning Port State for this port for frames with this VID. 0x3 = 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 69: VTU/STU Data Register for VTU Ops
Offset: 0x09 or decimal 9 – G1

Bits	Field	Type	Description
15	VIDPRI Override	RWR	VID Priority Override. When this bit is set to a one the VIDPRI bits (below) can be 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 and the port's VTUPRIOverride is enabled (Port offset 0x0D).
11:0	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.

Table 70: ATU Control Register
Offset: 0x0A or decimal 10 – G1

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:4	AgeTime	RWS to 0x58	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 3.75 is the age time in seconds. For example: The default value of 0x58 is 88 decimal. 88 x 3.75 = 330 seconds or 5.5 minutes. The minimum age time is 0x1 or 3.75 seconds. The maximum age time is 0xFF or 956 seconds or almost 16 minutes. If the AgeTime is set to 0x00 the Aging function is disabled and all learned addresses will remain in the database forever.
3	Learn2All	RWR	Learn to All devices in a Switch. When more than one Marvell device is used to form a single 'switch' it may be desirable for all devices in the 'switch' to learn any address this device learns. When this bit is set to a one all other devices in the 'switch' learn the same addresses this device learns[1]. 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.
2	LearnAvb	RWR	Learn on AVB frames. When this bit is cleared to a zero AVB flows are not learned from and Source Address refreshed are not done on AVB frames – nor are the Learn2All messages sent for these flows. When this bit is set to a one all of the above is possible on AVB flows. Not learning nor refreshing Source Addresses on AVB flows is not necessary for normal network operation (as these operations will still occur on non-AVB frames) and enabling this function needs to be done with care – especially taking into account the additional AVB Class bandwidth of the extra AVB flows going out the MessagePorts due to Learn2All being enabled (above).
1:0	HashSel	RWS to 0x1	Hash Select. These two bits are used to select the ATU's Hash as follows: 0x0 = Reserved 0x1 = Default 0x2 = Reserved 0x3 = Direct – For Test purposes only

1. Learn2All message learning frames will be sent out a port if that port's MessagePort bit is set to a one (Port offset 0x05). If this feature is used it is recommended that all Marvell Tag ports, except for the CPU's port, have their MessagePort bit set to a one. Ports that are not Marvell Tag ports (i.e., normal Network ports) should not have their MessagePort bit set to a one.

Table 71: ATU Operation Register
Offset: 0x0B or decimal 11 – G1

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 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 (Global 1 offset 0x04).
14:12	ATUOp	RWR	Address Translation Unit Opcode. The device supports the following ATU operations (all of these operations can be executed while frames are transiting through the switch): 0x0 = No Operation 0x1 = Flush[1]/Move[2] All Entries 0x2 = Flush[3]/Move all Non-Static Entries 0x3 = Load[4] or Purge[5] an Entry in a particular FID Database 0x4 = Get Next[6] from a particular FID Database 0x5 = Flush/Move All Entries in a particular FID Database 0x6 = Flush/Move all Non-Static Entries in a particular FID Database 0x7 = Get/Clear Violation Data[7]
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 6.3) and the port's SA and/or DA priority overrides are enabled (Port offset 0x0D).
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, it DPV or Trunk ID and its Entry State (=0x01). The violating MAC's PRI bits will be updated in MACPri (Global offset 0x0B) and it BIN will be updated in Global offset 0x06).

Table 71: ATU Operation Register (Continued)
Offset: 0x0B or decimal 11 – G1

Bits	Field	Type	Description
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 was due to a Source Address look-up that resulted in a Hit but where the ATUData[9:0] bits did 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 1st Member Violation, Miss Violation (below) or Full Violation (below) will be saved per port until cleared.
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 was 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 halfway aged out). Only the 1st Miss Violation, Member Violation (above) or Full Violation (below) will be 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 was due to a Load ATUOp or automatic learn that could not store the desired entry. This will only occur if all available locations for the desired address contain other MAC addresses that are loaded Static. Only the 1st Full Violation, Member Violation (above) or Miss Violation (above) will be saved per port until cleared.
3:0	Reserved	RES	Reserved for future use.

1. A Flush occurs if the EntryState (Global 1 offset 0x0C) is zero.
2. Move is used for 802.1w (rapid spanning tree) to reassign all valid entries associated with one port (the FromPort - Global 1 offset 0x0C) and move the association to another port (the ToPort - Global 1 offset 0x0C). It can also be used to completely deassociated a port from the database (if the ToPort = 0xF). The Move occurs if the EntryState (Global 1 offset 0x0C) 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. An Entry is Loaded if the EntryState (Global 1 offset 0x0C) is non-zero.
5. An Entry is Purged if it exists and if the EntryState (Global 1 offset 0x0C) is zero.
6. 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 (Global 1 offset 0x0D) are used as the address to start from. To find the lowest MAC address set ATUByte[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 Entry_State of 0x0, no higher MAC address was found. If ATUByte[5:0] is returned set to all one's with a non-zero Entry_State, 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.
7. When the ATUProb bit is set to a one (Global 1 offset 0x00) 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 will be cleared to a zero.

Table 72: ATU Data Register
Offset: 0x0C or decimal 12 – G1

Bits	Field	Type	Description
15	LAG	RWR	Link Aggregation (LAG) Mapped Address. When this bit is set to a one the data in bits 7:4 below (PortVec bits [3:0]) is the LAG 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 9:4 below (PortVec bit [5:0]) is the port vector assigned to this address.
14:12	Reserved	RES	Reserved for future use.
11:4	PortVec/ ToPort & FromPort	RWR	Port Vector. 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 Remove Port (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 input Entry State for ATU Load/Purge or Flush/Move operations and it's the resulting Entry State from ATU Get Next operations. If these bits equal 0x0 then the ATUOp will be a Purge or a Flush. If these bits are not 0x0 then the ATUOp will be 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 bits return 0x0. 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 AVB/Non Rate Limiting (NRL) entry
- 0xB: Static AVB/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

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 AVB/Non Rate Limiting (NRL) entry
- 0x6: Static entry defining frames with this DA as MGMT
- 0x7: Static entry
- 0xC: Static Policy entry with Priority Override
- 0xD: Static AVB/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

Note The Audio Video Bridging (AVB) selections above are only valid for devices that support AVB/TSN.

Table 73: ATU MAC Address Register Bytes 0 & 1
Offset: 0x0D or decimal 13 – G1

Bits	Field	Type	Description
15:8	ATUByte0	RWR	ATU MAC Address Byte 0 (bits 47:40). Used as the input MAC address for ATU Load, Purge or Get Next operations and it's the resulting MAC address from ATU Get Next operations. Bit 0 of byte 0 (bit 40) is the multicast bit (it is the 1st 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's 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 74: Switch MAC Address Register Bytes 2 & 3
Offset: 0x0E or decimal 14 – G1

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's 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's 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 75: Switch MAC Address Register Bytes 4 & 5
Offset: 0x0F or decimal 15 – G1

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's 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's 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 76: IEEE-PRI Register
Offset: 0x18 or decimal 24 – G1

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 77: Mapping Table
Offset: 0x19 or decimal 25 – G1

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 IP Mapping register selected by the Pointer bits below (Reserved bits do not exist). After the write has taken place this bit self clears to zero.
14	UseIpFPri	RWR	Use IP Frame Priorities from this table. This bit is used to be maintain backwards compatibility. When this bit is cleared to a zero, the IP_FPRI data in this table is ignored. Instead the frame's initial IP_FPRI is generated by using the frame's IP_QPRI as the IP_FPRI's upper two bits, and the IP_FPRI's lowest bit comes from bit 0 of the frame's source port's Default PRI (Port offset 0x07). When this bit is set to a one, the IP_FPRI data in this table is used as the frame's initial IP_FPRI.
13:8	Pointer	RWR	Pointer to the desired entry of the IP Mapping table. These bits select one of 64 possible IP mapping registers for both read and write operations (but not all entries exist). A write operation occurs if the Update bit is a one. Otherwise a read operation occurs. When a frame is received on a port, and if its an Ipv4 frame, the frame's six DiffServ bits are used to access this table to determine the frame's initial IP_QPRI and its initial IP_FPRI, depending upon the settings of the port's InitialPri and TagIfBoth bits (Port offset 0x04). If the frame is a Ipv6 frame, the frame's six Traffic Class bits are used in the same way to access this table. The reset values in this table are as follows: Pointer Range — IP_QPRI — IP_FPRI 0x00 to 0x0F — 0x0 — 0x0 0x10 to 0x1F — 0x1 — 0x2 0x20 to 0x2F — 0x2 — 0x4 0x30 to 0x3F — 0x3 — 0x6
7	Reserved	RES	Reserved for future use.
6:4	IP_FPRI	RWS See text	IPv4 and IPv6 Frame Priority Mapping. The value in this field is used as the frame's initial FPRI when the frame is an IPv4 or an IPv6 frame, and the port's InitialPri (Port offset 0x04) is configured to use IP Fpri's.
3:2	Reserved	RES	Reserved for future use.

Table 77: Mapping Table
Offset: 0x19 or decimal 25 – G1

Bits	Field	Type	Description
1:0	IP_QPRI	RWS See text	IPv4 and IPv6 Queue Priority Mapping. The value in this field is used as the frame's initial QPRI when the frame is an IPv4 or an IPv6 frame, and the port's InitialPri and TagIfBoth registers (Port offset 0x04) are configured to use IP Qpri's. **NOTE:** These bits are also accessible using the IP-PRI Mapping registers (Global 1 offsets 0x10 to 0x0F). Both access methods are supported for backwards compatibility. But new software should use this register as the function of the Global 1 offset 0x10 to 0x0F registers will be re-defined in future devices.

Table 78: Monitor & MGMT Control Register
Offset: 0x1A or decimal 26 – G1

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 Monitor & MGMT octet register referenced 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 the desired octet of Monitor & MGMT. These bits select one of the possible Monitor & MGMT registers for both read and write operations. A write operation occurs if the Update bit is a one (the Monitor & MGMT registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the Monitor & MGMT register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 to 0x01 = Rsvd2Cpu Enables for 01:C2:80:00:00:0x 0x02 to 0x03 = Rsvd2Cpu Enables for 01:C2:80:00:00:2x 0x04 to 0x05 = Rsvd2IntCpu Enables for 01:C2:80:00:00:0x 0x06 to 0x07 = Rsvd2IntCpu Enables for 01:C2:80:00:00:2x 0x04 to 0x1F = Reserved for future use. 0x20 to 0x2F = Mirror Destination Port Settings 0x30 to 0x3F = Trap Destination Port Settings
7:0	Data	RWR	Octet Data of the Monitor & MGMT register. As referenced by the Pointer bits above.

The individual registers accessed by the Port Queue Control register are described below and are indicated with a light purple background heading color.

Table 79: Watch Dog Interrupt Source, Index: 0x00 of Watch Dog – G2 Offset 0x1B

Bits	Field	Type	Description
7:0	Rsvd2Cpu Enables 0x Low Byte	RWR	Reserved DA's to CPU Enables 0x – Low Byte. These bits represent the lower 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with that reserved multicast DA address are treated as MGMT[1] frames and are trapped to the port pointed to by CPUDest (Index 0x30 below). 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 = 0x7 bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the CPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Table 80: Rsvd2Cpu Byte 1, Index: 0x01 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2Cpu Enables 0x High Byte	RWR	Reserved DA to CPU Enables 0x – High Byte. These bits represent the upper 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with that reserved multicast DA address are treated as MGMT[1] frames and are trapped to the port pointed to by CPUDest (Index 0x30 below). All the reserved DA's supported by this register take the form 01:80:C2:00:00:0x. When x = 0x8, bit 0 of this register is tested. When x = 0xA bit 2 of this field is tested and so on with x = 0xF bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the CPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Note: The same frame can not be trapped to both the CPUDest and the AltCPUDest (see indexes 0x04 to 0x07). CPUDest will win between the two. If a given address needs to be processed by both CPU's send it to one of them and have the 1st CPU send a copy to the 2nd CPU.

Table 81: Rsvd2Cpu Byte 2, Index: 0x00 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2Cpu Enables 2x Low Byte	RWR	Reserved DA to CPU Enables 2x – Low Byte. These bits represent the lower 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with the reserved multicast DA address associated with that bit are treated as MGMT[1] frames and are trapped to the port pointed to by CPUDest (Index 0x30 below). All 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 = 0x7 bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the CPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Table 82: Rsvd2Cpu Byte 3, Index: 0x01 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2Cpu Enables 2x High Byte	RWR	Reserved DA to CPU Enables 2x – High Byte. These bits represent the upper 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with the reserved multicast DA address associated with that bit are treated as MGMT[1] frames and are trapped to the port pointed to by CPUDest (Index 0x30 below). All the reserved DA's supported by this register take the form 01:80:C2:00:00:2x. When x = 0x8, bit 0 of this register is tested. When x = 0xA bit 2 of this field is tested and so on with x = 0xF bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the CPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Note: The same frame can not be trapped to both the CPUDest and the AltCPUDest ((see indexes 0x04 to 0x07). CPUDest will win between the two. If a given address needs to be processed by both CPU's send it to one of them and have the 1st CPU send a copy to the 2nd CPU.

Table 83: Rsvd2AltCpu Byte 0, Index: 0x04 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2 AltCpu Enables 0x Low Byte	RWR	Reserved DA to Alternate CPU Enables 0x – Low Byte. These bits represent the lower 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with the reserved multicast DA address associated with that bit are treated as MGMT[1] frames and are trapped to the port pointed to by AltCPUDest (Index 0x31 below). 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 = 0x7 bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the AltCPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Table 84: Rsvd2AltCpu Byte 1, Index: 0x05 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2 AltCpu Enables 0x High Byte	RWR	Reserved DA to Alternate CPU Enables 0x – High Byte. These bits represent the upper 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with the reserved multicast DA address associated with that bit are treated as MGMT[1] frames and are trapped to the port pointed to by AltCPUDest (Index 0x31 below). All the reserved DA's supported by this register take the form 01:80:C2:00:00:0x. When x = 0x8, bit 0 of this register is tested. When x = 0xA bit 2 of this field is tested and so on with x = 0xF bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the AltCPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Note: The same frame can be trapped to both the CPUDest (see indexes 0x00 to 0x03) and the AltCPUDest. CPUDest will win between the two. If a given address needs to be processed by both CPU's send it to one of them and have the 1st CPU send a copy to the 2nd CPU.

Table 85: Rsvd2AltCpu Byte 2, Index: 0x06 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2 AltCpu Enables 2x Low Byte	RWR	Reserved DA to Alternate CPU Enables 2x – Low Byte. These bits represent the lower 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with the reserved multicast DA address associated with that bit are treated as MGMT[1] frames and are trapped to the port pointed to by AltCPUDest (Index 0x31 below). All 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 = 0x7 bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the AltCPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Table 86: Rsvd2AltCpu Byte 3, Index: 0x07 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:0	Rsvd2 AltCpu Enables 2x High Byte	RWR	Reserved DA to Alternate CPU Enables 2x – High Byte. These bits represent the upper 8 of 16 reserved multicast DA addresses. When a bit in this register is set to a one, frames with the reserved multicast DA address associated with that bit are treated as MGMT[1] frames and are trapped to the port pointed to by AltCPUDest (Index 0x31 below). All the reserved DA's supported by this register take the form 01:80:C2:00:00:2x. When x = 0x8, bit 0 of this register is tested. When x = 0xA bit 2 of this field is tested and so on with x = 0xF bit 7 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 as far as this function is concerned This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames that are trapped to the AltCPUDest.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <???MGMT Frame Handling>).

Note: The same frame can not be trapped to both the CPUDest (see indexes 0x00 to 0x03) and the AltCPUDest. CPUDest will win between the two. If a given address needs to be processed by both CPU's send it to one of them and have the 1st CPU send a copy to the 2nd CPU.

Table 87: Ingress Monitor Dest, Index: 0x20 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:5	Reserved	RES	Reserved for future use.
4:0	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 function is disabled when this register is set to 0x1F. This includes frames received on a Marvell Tag port with the Ingress Monitor type, and frames received on a Network port that is enabled to be the Ingress Monitor Source Port (Port offset 0x08). 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 Marvell Tag port in this device that is used to get to the device that contains the Ingress Monitor Destination Port. Use a value of 0x06 to map the frames to the internal CPU.

Table 88: Egress Monitor Dest, Index: 0x21 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:5	Reserved	RES	Reserved for future use.
4:0	Egress Monitor Dest	RWS	Egress Monitor Destination Port. Frames that are targeted toward an Egress Monitor Destination go out the port number indicated in these bits. This function is disabled when this register is set to 0x1F. This includes frames received on a Marvell Tag port with the Egress Monitor type, and frames transmitted on a Network port that is enabled to be the Egress Monitor Source Port (Port offset 0x08). 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 Marvell Tag port in this device that is used to get to the device that contains the Egress Monitor Destination Port. Use a value of 0x06 to map the frames to the internal CPU.

Table 89: Mirror Dest, Index: 0x22 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:5	Reserved	RES	Reserved for future use.
4:0	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. This function is disabled when this register is set to 0x1F. 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 = 0x1F Policy Mirroring is disabled and ingressing To_CPU Policy Mirror frames will be discarded. Use a value of 0x06 to map the frames to the internal CPU. The policy mirror enable bits are configurable per port (see Policy Control, Port offset 0x0E, and Miss Mirrors, Port offset 0x0D).

Table 90: CPU Dest, Index: 0x30 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:5	MgmtPri	RWS to 0x7	MGMT Priority. These bits are used as the priority to use on Rsvd2CPU frames (Index 0x00 to 0x03 above).
4:0	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 on the port (Port offset 0x08) 7. When any of the port's Miss Traps occur that are enabled (Port offset 0x0D) 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. These functions are disabled when this register is set to 0x1F. 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 in the above list). 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 = 0x1F the remapped frames will be discarded, no ARP mirroring will occur and ingressing To_CPU frames will be discarded. Use a value of 0x06 to map the frames to the internal CPU. **NOTE:** MGMT or BPDU frames detected by using the ATU are directed to the correct port where the CPU is connected by insuring the CPU port's bit is set in the frame's MGMT DA MAC address as stored in the ATU address database (section <???ATU Entry Format>).

1. To_CPU frames with a Code of Policy Mirror are mapped to the MirrorDest port (Index 0x22 above).

Table 91: Alternate CPU Dest, Index: 0x31 of Monitor & MGMT Control – G1 Offset 0x1A

Bits	Field	Type	Description
7:5	AltPri	RWS to 0x7	Alternate MGMT Priority. These bits are used as the priority to use on Rsvd2AltCPU frames (Index 0x04 to 0x07 above).
4:0	AltCPU Dest	RWS	Alternate CPU Destination Port. Frames that ingress a port that match the Rsvd2AltCpu criteria defined above (Index 0x04 to 0x07) are treated as MGMT[1] and mapped to this port only (trapped) overriding where the frame would normally go. This function is disabled when this register is set to 0x1F. The AltCPUDest bits indicate the port number on this device where the alternate CPU is connected (either directly or indirectly through another Marvell switch device – the next switch device must also have the same Rsvd2AltCPU criteria as this device for this to work). Use a value of 0x06 to map the frames to the internal CPU.

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see section <MGMT Frame Handling>).

Table 92: Total Free Counter
Offset: 0x1B or decimal 27 – G1

Bits	Field	Type	Description
15:10	Reserved	RES	Reserved for future use.
9:0	FreeQSize	RO	Free Queue Size Counter. This counter reflects the current number of unallocated buffers available for all the ports.

Table 93: Global Control 2
Offset: 0x1C or decimal 28 – G1

Bits	Field	Type	Description
15:14	Header Type	RWR	Header Type These bits are used to configure the bits that are placed into the Egress Header when it is enabled on a port (Port offset 0x04) as follows: 0x0 = Original Header – for backwards compatibility to UniMAC's that look at Header byte 1 bits[4:2] and byte 2 bits [3:0] 0x1 = Single chip MGMT Header – for compatibility to Marvell Fast Ethernet switches that support Spanning Tree without DSA Tags 0x2 = Trunk Header – used together with the DSA Tags to perform Remote Switching 0x3 = Reserved for future use. See section ??? on the exact content of each of these Headers.
13:12	Reserved	RES	Reserved for future use.
11	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.

Table 93: Global Control 2 (Continued)
Offset: 0x1C or decimal 28 – G1

Bits	Field	Type	Description
10:8	RMUMode A value of 0x6 is not valid for 88E6141	RWS to 0x7	Remote Management Unit Mode 0x0 = Port 0 is enabled to be the RMU (Remote Management Unit) port for the switch. 0x1 = Port 1 is enabled to be the RMU port for the switch. 0x2 = Port 5 is enabled to be the RMU port for the switch. 0x3 to 0x5 = Reserved for future use. 0x6 = All DSA or EtherType DSA ports can receive RMU frames which are sent to the internal CPU for processing. 0x7 = RMU feature is disabled. When RMU is enabled and this device receives a Remote Management Request frame directed to this device the frame will be processed and a Remote Management Response frame will be generated and sent out if the DA of the frame matches the conditions of the DA Check bit above. In all cases, the Request frame will be discarded (as it was directed to this device) or it will be mapped to the internal CPU for processing. When RMU is disabled, 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 these bits, Remote Management Request frames that are not directed to this device will be mapped to the port indicated by mapping the frame's Trg_Dev using the Device Mapping table (Global 2 offset 0x06). **NOTE:** The setting of these bits will have no effect if the Remote Management port is in half duplex mode. The ingress port's FrameMode (Port offset 0x04) must be DSA or EtherType DSA as well.
7:6	Histogram Mode	RWR	Histogram Counters Mode. The Histogram mode bits control how the Histogram counters accessed via Global 1 offset 0x1D work as follows: 0x0 = Reserved 0x1 = Count received frames only 0x2 = Count transmitted frames only 0x3 = Count receive and transmitted frames

Table 93: Global Control 2 (Continued)
Offset: 0x1C or decimal 28 – G1

Bits	Field	Type	Description
5	CtrMode	RWR	Counter Mode. This bit controls the operating mode of the Port's Debug counter at Port offset 0x1F. When CtrMode is cleared to a zero the Debug counter for all ports (Port offset 0x1F) counts RxBad frames in the upper 8 bits of the register and counts RxGood frames in the lower 8 bits of the register. When this bit is set to a one the Debug counter for all ports counts Collisions in the upper 8 bits of the register and counts Tx Transmitted frames in the lower 8 bits of the register. The Debug Counters for all ports are cleared to a zero whenever this bit changes state (i.e., it transitions from a one to a zero or from a zero to a one).
4:0	Device Number	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 Trg_Dev field matches these bits have reached their destination chip and are sent out this chip using the port number indicated in the frame's Trg_Port field. The DeviceNumber bits 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[3:0] configuration pins are used to set the initial value of this register (ADDR[4] is always zero in this device). The ADDR[3:0] pins are also used to select between Multi Chip addressing mode or Single Chip addressing mode. Changing the value in this register after reset will *not* change the device's addressing mode.

Table 94: Stats Operation Register
Offset: 0x1D or decimal 29 – G1

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 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 (Global 1 offset 0x04).
14:12	StatsOp	RWR	Statistics Unit Opcode. The device supports the following Stats operations (all of these operations can be executed while frames are transiting through the switch): 0x0 = No Operation 0x1 = Flush (clear) All Counters for all Ports 0x2 = Flush (clear) All Counters for a Port 0x3 = Reserved 0x4 = Read a Captured or Direct Counter 0x5 = Capture All Counters for a Port 0x6 = Reserved 0x7 = Reserved
11	Reserved	RES	Reserved for future use.
10	StatsBank	RWR	Statistics Bank of Counters. When this bit is cleared to a zero the MAC based MIBs (Bank 0) are accessed when a 'Read a Captured or Direct Counter' StatsOp is performed. When this bit is set to a one the Policy based MIBs (Bank 1) are accessed when a 'Read a Captured or Direct Counter' StatsOp is performed. The value of this register has no effect on the other StatsOps commands as each StatsOp command is done to both MIB Banks at the same time.
9	Reserved	RES	Reserved for future use.
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. Set these bits = 0x0 to access the captured counters. Set these bits = 0x1 to access the counters for Port 0. Set these bits = 0x2 to access the counters for Port 1, etc. These bits represent the port number for the following StatOps: a. Flush (clear) All Counters for a Port b. Read a Captured or Direct Counter c. Capture All Counters for a Port

Table 94: Stats Operation Register
Offset: 0x1D or decimal 29 – G1

Bits	Field	Type	Description
4:0	StatsPtr	RWR	Statistics Pointer. This field is used as a parameter for the above StatsOp commands. StatsPtr must be set to the desired counter to read for the Read a Captured or Direct 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 with StatsPort = 0x0. The counter that is read is defined as follows: **Bank 0 Ingress Counters**[1] 0x00 – InGoodOctetsLo 0x01 – InGoodOctetsHi 0x02 – InBadOctets 0x04 – InUnicast 0x06 – InBroadcasts 0x07 – InMulticasts 0x16 – InPause 0x18 – InUndersize 0x19 – InFragments 0x1A – InOversize 0x1B – InJabber 0x1C – InRxErr 0x1D – InFCSErr 0x1F – Late **Bank 0 Egress Counters** 0x0E – OutOctetsLo[2] 0x0F – OutOctetsHi 0x10 – OutUnicast 0x13 – OutBroadcasts 0x12 – OutMulticasts 0x15 – OutPause 0x1E – Collisions 0x05 – Deferred 0x14 – Single 0x17 – Multiple 0x03 – OutFCSErr 0x11 – Excessive **Bank 0 Histogram Counters**[3] 0x08 – 64Octets 0x09 – 65to127Octets 0x0A – 128to255Octets 0x0B – 256to511Octets 0x0C – 512to1023Octets 0x0D – 1024toMaxOctets **Bank 1 Ingress Counters** 0x00 – InDiscards 0x01 – InFiltered 0x02 – InAccepted 0x03 – InBadAccepted 0x04 – InGoodAvbClassA 0x05 – InGoodAvbClassB 0x06 – InBadAvbClassA 0x07 – InBadAvbClassB 0x08 – TCAMCounter0 0x09 – TCAMCounter1 0x0A – TCAMCounter2 0x0B – TCAMCounter3 0x0C – InDroppedAvbA 0x0D – InDroppedAvbB 0x0E – InDaUnknown 0x0F – InMGMT **Bank 1 Egress Counters** 0x10 – OutQueue0 0x11 – OutQueue1 0x12 – OutQueue2 0x13 – OutQueue3 0x14 – OutQueue4 0x15 – OutQueue5 0x16 – OutQueue6 0x17 – OutQueue7 0x18 – OutCutThrough 0x19 – Reserved 0x1A – OutOctetsA 0x1B – OutOctetsB 0x1C – OutYellow 0x1D – OutDroppedYel 0x1E – OutDiscards 0x1F – OutMGMT

1. If Marvell Header mode is used 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 transmited on frames that encounter a collision.

3. If Marvell Header mode is used the extra two bytes in the frame are *not* included in the count before determining which Histogram Counter to increment.

Table 95: Stats Counter Register Bytes 3 & 2
Offset: 0x1E or decimal 30 – G1

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 – Global 1 offset 0x1D). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.
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 – Global 1 offset 0x1D). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.

Table 96: Stats Counter Register Bytes 1 & 0
Offset: 0x1F or decimal 31 – G1

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 – Global 1 offset 0x1D). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.
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 – Global 1 offset 0x1D). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.

1.3.3 Switch Global 2 Registers

The device contains 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 assignments are shown in Table 97.

Table 97: Table 5: Global 2 Register bit Map (Device Addr 0x1C)

Global 2 Register Data Bits

SMI Register Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	WD	JL	DM	WE	Reserved						SI	PHY Ints				R	0 Interrupt Source
1	WD	JL	DM	WE	Reserved						SM	PHY Int Masks				R	1 Interrupt Masks
2	Reserved																2
3	Reserved																3
4	U	SPD			Flow Control Delay TIme												4 Flow Control Delays
5	RES		FM	FB	R1	AI	TF	AD	FF	FC Pri				Reserved			5 Managment
6	U	RES		Trg_Dev Value					Reserved			Trg_Dev Port					6 Device Mapping
7	U	Mask Num		H	Reserved				LAG Mask								7 LAG Mask
8	U	LAG ID			Reserved				LAG Map								8 LAG Mapping
9	IB	IRL Op		IRL Port				IRL Res		R	IRL Reg						9 Ingress Rate Command
A	Ingress Rate Data																10
B	PB	PVTOp		Reserved			Pointer										11 Cross Chip Port VLAN Addr
C	Reserved							PVLAN Data									12 Cross Chip Port VLAN Data
D	U	RES		Pointer				Data									13 Switch MAC/WoL/WoF
E	Bin		Count Mode	Active ATU Bin Entries													14 ATU Stats
F	U	FS	RES	Pointer			AO		Data AVB	OE	Data						15 Priority Overide Table
10	U	Pointer			Data												16 I2C Block
11	Reserved																17
12	U	Pointer					Data										18 Energy Management
13	U	Pointer					Data										19 IMP Comm/Debug
14	B	EEOp		ER	WE	RES	EEPROM Data										20 EEPROM Command
15	EEPROM Address																21 EEPROM Address
16	B	AVBOp		AVBPort			AVBBlock		AVBAddr								22 AVB Command
17	AVBData																23 AVB Data
18	B	SMI Mode		M	SMIOp		DevAddr		RegAddr								24 SMI PHY Command
19	SMI Data																25 SMI PHY Data
1A	U	Pointer					Data										26 Scracth & Misc
1B	U	Pointer					Data										27 Watch Dog Control
1C	U		Pointer				Data										28 QoS Weights
1D		5B	NE	Reserved													29 Misc
1E	Reserved																30
1F	RES			Cut Through Hold				Reserved									31 Cut Through Control
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

WD = Watch Dog Int
JL = Jam Limit Int
DM = Duplex Mismatch
WE = Wake Event (Wol & WoF)
SI = SERDES Int
SM = SERDES Int Mask

U = Update

FM = Flow Control Messages
FB = Forward Broadcasts
R1 = Remove 1 Ptag
AI = ATU Age Interrupt
TF = Use Tag FC Data
AD = Auto Disable on Link Down
FF = Force FC Priority

U = Update

U = Update
H = Hask LAG

U = Update

IB = IRL Busy

PB = PVT Busy

U = Update

U = Update
FS = FPriSet of Registers
AO = AVB QPri Override En
OE = QPri Override En

U = Update

U = Update

U = Update

B = Busy
ER = EEPROM Running
WE = Write Enable

B = Busy

B = Busy

U = Update

U = Update

U = Update

5B = 5 Bit Port
NE = No Egress Policy

Table 98: Interrupt Source Register
Offset: 0x00 or decimal 0 – G2

Bits	Field	Type	Description
15	WatchDog Int	RO	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	Duplex Mismatch	ROC	Duplex Mismatch interrupt. This bit is set to a one when any of the ports detect a duplex mismatch (i.e., the local port is in half duplex mode while the link partner is in full duplex mode). When this bit is set to a one, the port that the duplex mismatch was detected on will be seen in the DMPort register (Global 2 offset 0x1B, index 0x40). Software must set the DMPort register back to a value of 0xF to re-arm the Duplex Mismatch interrupt.
12	WakeEvent	RO	Wake Event interrupt. This bit is set to a one when any of the ports detect an enabled Wake Event such as Wake on LAN or Wake on Frame. When this bit is a one, software can find the source of the interrupt by reading the WoF Int bit and the WoL Port bits (both registers are in index 0x1F of Switch MAC/WoL/WoF at Global 2 offset 0x0D).
11:6	Reserved	RES	Reserved for future use.
5	SERDESInt	RO	SERDES layer core interrupt bit. This bit is set when any of Port 5's internal SERDES core's interrupt bits are set. A port's SERDESInt bit will clear to zero when all the unmasked interrupts from the port's SERDES are serviced.
4:1	PHYInt	RO	PHY layer core interrupt bit. This bit is set when any of the internal PHY core's interrupt bits are set. A port's PHYInt bit will clear to zero when all the unmasked interrupts from the port's PHY are serviced. Bit 4 is for port 4, bit 3 is for port 3 etc.
0	Reserved	RES	Reserved for future use.

Table 99: Interrupt Mask Register
Offset: 0x01 or decimal 1 – G2

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	RWR	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	Duplex Mismatch En	RWR	Duplex Mismatch interrupt enable. This bit must be set to a one to allow the Duplex Mismatch 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.
12	WakeEvent En	RWR	Wake Event interrupt enable. This bit must be set to a one to allow the WakeEvent 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.
11:6	Reserved	RES	Reserved for future use.
5	SERDES IntEn	RWR	SERDES layer core interrupt enable bit. This bit is set to a one to allow SERDES Interrupts from Port 5 to drive the DeviceInt bit in the Switch Global Status register (Global offset 0x00) so that the INTn pin can be driven low.
4:1	PHYIntEn	RWR	PHY layer core interrupt enable bit. This bit is set to a one to allow PHY Interrupts from a given physical layer core to drive the DeviceInt bit in the Switch Global Status register (Global offset 0x00) so that the INTn pin can be driven low. Bit 4 is for port 4, bit 3 is for port 3 etc.
0	Reserved	RES	Reserved for future use.

Table 100: Flow Control Delay Register
Offset: 0x04 or decimal 4 – G2

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 Marvell 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 value for each of these registers is shown below. SPD 0b00 resets to 0x0258 (600 decimal) SPD 0b01 resets to 0x003C (60 decimal) SPD 0b10 resets to 0x0006 (6 decimal)

Table 101: Switch Management Register
Offset: 0x05 or decimal 5 – G2

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13	Flow Control Message	RWR	Enable Flow Control Messages[1]. When this bit is set to a one Marvell Tag Flow Control messages will be generated when a Flow Control enabled output queue becomes congested. When this bit is cleared to a zero Marvell 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 Default Forward 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).

Table 101: Switch Management Register
Offset: 0x05 or decimal 5 – G2

Bits	Field	Type	Description
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 halfway 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.
9	Tag Flow Control	RWR	Tag Flow Control. 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 cleared 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	Auto Disable	RWR	Auto Disable on Link Down. When this bit is set to a one and a port's Link (as seen in the port's Link register – Port offset 0x00) goes down, the port's Port State (Port offset 0x04) will be set to the Disabled Port State. This ensures that no packets can ingress or egress this port once Link is re-established until software is aware of the new link up. When this bit is cleared to a zero the port's Port State bits will not be modified on Link down.
7	ForceFlow ControlPri	RWS	Force Flow Control Priority. When this bit is set to a one the PRI[2:0] bits of generated Marvell 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 Marvell 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 Marvell Tag Flow Control frames if the ForceFlowControlPri bit above is set to a one.
3:0	Reserved	RES	Reserved for future use.

1. 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 102: Device Mapping Table Register
Offset: 0x06 or decimal 6 – G2

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. 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 the Trg_DevValue is connected. In this way a physical mapping of the interconnection of the devices that make up the switch can be defined. When a write occurs to this register with the Update bit being a one these bits are written to the Trg_DevPort selected by the Trg_DevValue 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_DevValue 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_DevValue bits. This table is reset to the following values: Trg_Dev Value Trg_Dev Port Trg_Dev Value Trg_Dev Port 0x00 0xF 0x10 0xF 0x01 0xF 0x11 0xF 0x02 0xF 0x12 0xF 0x03 0xF 0x13 0xF 0x04 0xF 0x14 0xF 0x05 0xF 0x15 0xF 0x06 0xF 0x16 0xF 0x07 0xF 0x17 0xF 0x08 0xF 0x18 0xF 0x09 0xF 0x19 0xF 0x0A 0xF 0x1A 0xF 0x0B 0xF 0x1B 0xF 0x0C 0xF 0x1C 0xF 0x0D 0xF 0x1D 0xF 0x0E 0xF 0x1E 0xF 0x0F 0xF 0x1F 0xF

Table 103: LAG Mask Table Register
Offset: 0x07 or decimal 7 – G2

Bits	Field	Type	Description
15	Update	SC	Update LAG (Link Aggregation) Mask data. When this bit is set to a one the data written to bits 5:0 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	Mask Num	RWR	Mask Number. These bits select one of eight possible LAG (Link Aggregation) Mask vectors for both read and write operations. A write operation occurs if the Update bit is a one (the LAG Mask Table can be written by writing to this register once).Otherwise a read of the current MaskNum occurs and the data found there is placed in the LAG Mask bits below (the LAG Mask Table can be read by first writing to this register, with Update = 0, and then reading this register).
11	Hash LAG	RWR	Hash DA & SA for LAG Mask selection. LAG (Link Aggregation) load balancing is accomplished by using the frame's DA and SA fields to access one of eight LAG Masks, unless the TCAM's Load Balance Override is set. When this bit is set to a one the hash computed for DA & SA address table lookups is used for the LAGMask 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 LAGMask to use. When the TCAM's Load Balance Override is set on a frame, this bit has no effect on that frame as the TCAM's Load Balance Data is used instead
10:7	Reserved	RES	Reserved for future use.
6:0	LAG Mask	RWS	LAG (Link Aggregation) 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 LAG 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 LAGNum bits to be written to for read operations). When a read occurs to this register these bits reflect the LAG Mask data found for the entry selected by the LAGNum bits. The LAGMask is reset to all ones for all LAGNum entries.

Trunk Mapping Register, Offset: 0x08 or decimal 8 – G2

Bits	Field	Type	Description
15	Update	SC	Update LAG (Link Aggregation) Mapping data. When this bit is set to a one the data written to bits 10:0 will be loaded into the LAG Map selected by the LAG ID bits below. After the write has taken place this bit self clears to zero.
14:11	LAG ID	RWR	LAG (Link Aggregation) Identifier. These bits select one of sixteen possible LAG 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.
10:7	Reserved	RES	Reserved for future use.
6:0	LAG Map	RWR	LAG (Link Aggregation) Map 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 LAG Map selected by the LAG 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 LAG ID bits to be written to for read operations). When a read occurs to this register these bits reflect the LAG Mapping data found for the entry selected by the LAG ID bits.

Table 104: Ingress Rate Command Register
Offset: 0x09 or decimal 9 – G2

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): 0x0 = No Operation 0x1 = Init all resources to the initial state i.e., all PIRL functions are disabled. 0x2 = Init the selected resource (pointed to by IRLPort, IRLRes and IRLReg) to the initial state i.e., selected rate resource is disabled. This initializes internal rate limiting related counters. 0x3 = Write to the selected resource/register (IRLPort/IRLRes/IRLReg)[1] 0x4 = Read the selected resource/register (IRLPort/IRLRes/IRLReg)[2] 0x5 = Reserved 0x6 = Reserved 0x7 = Reserved
11:8	IRLPort	RWR	Ingress Rate Limiting Port. These bits indicate the ingress rate limiting port that is being accessed. For example, if this field is programmed to a 0x2, it indicates that ingress rate resource belonging to port number 2 is being accessed.
7:5	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, 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 IRLPort and IRLRes above. Use a value of 0x0 to access register 0, a value of 0x1 to access register 1, etc.

1. 1st write the data to the IRLData register (Global 2, offset 0x0A), and then perform an IRLWrite
2. 1st perform an IRLRead and then the requested data will be available in the IRLData register (Global 2, offset 0x0A)

Table 105: Ingress Rate Data Register
Offset: 0x0A or decimal 10 – G2

Bits	Field	Type	Description
15:0	IRLData	RWR	Ingress Rate Limit Data. These data bits are either the read data or the write data bits depending on the IRL Command register (Global 2 offset 0x09). In the case of a read operation, the hardware logic fetches the data bits from the specified address in the IRL Command register and stores them into these bits. In the case of a write operation, the hardware logic utilizes the data bits in this field to write to the specified address location in the IRL Command register. The content of the IRL registers is documented in Section 1.3.4, Ingress Rate Limiter (IRL) Registers, on page 212.

Table 106: Cross Chip Port VLAN Addr Register
Offset: 0x0B or decimal 11 – G2

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): 0x0 = No Operation 0x1 = Init the PVT Table to all one's (initial state) 0x2 = Reserved 0x3 = Write PVLAN Data (Global 2 offset 0x0C) to the selected register[1] 0x4 = Read the selected register[2] 0x5 = Reserved 0x6 = Reserved 0x7 = 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 107: Cross Chip Port VLAN Data Register
Offset: 0x0C or decimal 12 – G2

Bits	Field	Type	Description
15:7	Reserved	RES	Reserved for future use.
6:0	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 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 1 entry of the 512 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 108: Switch MAC/WoL/WoF register
Offset: 0x0D or decimal 13 – G2

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 Switch MAC/WoL/WoF octet register referenced by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:13	Reserved	RES	Reserved for future use.
12:8	Pointer	RWR	Pointer to the desired octet of Switch MAC/WoL/WoF. These bits select one of the possible Switch MAC, Wake on Frame or Wake on LAN (WoL) registers for both read and write operations. A write operation occurs if the Update bit is a one (the Switch MAC, WoF or WoL registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the Switch MAC, WoF or WoL register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 to 0x05 = Switch MAC register space (used in Pause frames) 0x06 to 0x0B = Reserved for future use 0x0C to 0x0F = Wake on Frame (WoF) register space 0x10 to 0x1F = Wake on LAN (WoL) register space
7:0	Data	RWR	Octet Data of the Switch MAC/WoL/WoF register referenced by the Pointer bits above.

The individual registers accessed by the Port Queue Control register are described below and are indicated with a light purple background heading color.

Table 109: Switch MAC Byte 0, Index: 0x00 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:1	SwMAC Byte 0	RWR	Switch MAC Address Byte 0 (bits 47:41). Used as the switch's source address (SA) in transmitted full-duplex Pause frames. 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.
0	DiffAddr	RWR	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 bit = 0, all ports transmit the same SA. When this bit = 1, each port uses a different SA where bit 47:4 of the MAC address are the same, but bit 3:0 are the port number (Port 0 = 0, Port 1 = 1, etc.)

Table 110: Switch MAC Byte 1, Index: 0x01 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	SwMAC Byte 1	RWR	Switch MAC Address Byte 1 (bit 39:32). Used as the switch's source address (SA) in transmitted full-duplex Pause frames.

Table 111: Switch MAC Byte 2, Index: 0x02 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	SwMAC Byte 2	RWR	Switch MAC Address Byte 2 (bit 31:24). Used as the switch's source address (SA) in transmitted full-duplex Pause frames.

Table 112: Switch MAC Byte 3, Index: 0x03 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	SwMAC Byte 3	RWR	Switch MAC Address Byte 3 (bit 23:16). Used as the switch's source address (SA) in transmitted full-duplex Pause frames.

Table 113: Switch MAC Byte 4, Index: 0x04 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	SwMAC Byte 4	RWR	Switch MAC Address Byte 4 (bit 15:8). Used as the switch's source address (SA) in transmitted full-duplex Pause frames.

Switch MAC Byte 5, Index: 0x05 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	SwMAC Byte 5	RWR	Switch MAC Address Byte 5 (bit 7:0). Used as the switch's source address (SA) in transmitted full-duplex Pause frames.

Table 114: Wake on Frame Int, Index: 0x0C of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6	WoF Int P6	RO	Port 6 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 6's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P6 bit (index 0x0E of Switch MAC/WoL/WoF).
5	WoF Int P5	RO	Port 5 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 5's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P5 bit (index 0x0E of Switch MAC/WoL/WoF).
4	WoF Int P4	RO	Port 4 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 4's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P4 bit (index 0x0E of Switch MAC/WoL/WoF).
3	WoF Int P3	RO	Port 3 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 3's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P3 bit (index 0x0E of Switch MAC/WoL/WoF).
2	WoF Int P2	RO	Port 2 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 2's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P2 bit (index 0x0E of Switch MAC/WoL/WoF).
1	WoF Int P1	RO	Port 1 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 1's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P1 bit (index 0x0E of Switch MAC/WoL/WoF).
0	WoF Int P0	RO	Port 0 Wake on Frame Interrupt. When this bit is set to a one, a frame is pending in Port 0's egress queue. This bit will clear to a zero when the pending frame is released which is done by clearing to zero the WoF Int En P0 bit (index 0x0E of Switch MAC/WoL/WoF).

Table 115: Wake on Frame Enable, Index: 0x0E of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6	WoF Int En P6	RWR	Port 6 Wake on Frame Interrupt Enable. See the description for bit 0 below.
5	WoF Int En P5	RWR	Port 5 Wake on Frame Interrupt Enable. See the description for bit 0 below.
4	WoF Int En P4	RWR	Port 4 Wake on Frame Interrupt Enable. See the description for bit 0 below.
3	WoF Int En P3	RWR	Port 3 Wake on Frame Interrupt Enable. See the description for bit 0 below.
2	WoF Int En P2	RWR	Port 2 Wake on Frame Interrupt Enable. See the description for bit 0 below.
1	WoF Int En P1	RWR	Port 1 Wake on Frame Interrupt Enable. See the description for bit 0 below.
0	WoF Int En P0	RWR	Port 0 Wake on Frame Interrupt Enable. When this bit is set to a one, any frame that is mapped to Port 0's egress queue will be held (i.e., not transmitted), the Wake Event interrupt status will be set to a one (Global 2 offset 0x00), and the WoF Int P0 bit will be set to a one (Switch MAC/WoL/WoF offset 0x0C) causing the WoF Int to be set to a one (Switch MAC/WoL/WoF offset 0x1F). If the Wake Event Mask is set to a one (Global 2 offset 0x01) the device's INTn pin will be active (low). When this bit is cleared to a zero, all frames held in the port's egress queue will be allowed to transmit.

Table 116: WoL MAC Byte 0, Index: 0x10 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:1	WoL MAC Byte 0	RWR	Wake on LAN MAC Address Byte 0 (bits 47:41). Used as the WoL address to match n number of times as defined in the port's WoL Control registers (index 0x1C and 0x1D of Switch MAC/WoL/WoF). Since bit 0 of byte 0 (bit 40) is the multicast bit (it is the 1st bit down the wire) it is assumed as a zero unless set to a one in the WoL Control Byte 2 (index 0x1E of Switch MAC/WoL/WoF).
0	WoL DiffAddr	RWR	Wake on LAN Different MAC addresses per Port. This bit is used to have all ports match the same or different WoL addresses. When this bit = 0, all ports will match the same WoL address. When this bit = 1, each port matches a different WoL address where bit 47:4 of the MAC address are the same, but bit 3:0 are the port number (Port 0 = 0, Port 1 = 1, etc.)

Table 117: WoL MAC Byte 1, Index: 0x11 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL MAC Byte 1	RWR	Wake on LAN MAC Address Byte 1 (bit 39:32). Used as the WoL address to match n number of times (see description of WoL MAC Byte 0 above).

Table 118: WoL MAC Byte 2, Index: 0x12 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL MAC Byte 2	RWR	Wake on LAN MAC Address Byte 2 (bit 31:24). Used as the WoL address to match n number of times (see description of WoL MAC Byte 0 above).

Table 119: WoL MAC Byte 3, Index: 0x13 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL MAC Byte 3	RWR	Wake on LAN MAC Address Byte 3 (bit 23:16). Used as the WoL address to match n number of times (see description of WoL MAC Byte 0 above).

Table 120: WoL MAC Byte 4, Index: 0x14 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL MAC Byte 4	RWR	Wake on LAN MAC Address Byte 4 (bit 15:8). Used as the WoL address to match n number of times (see description of WoL MAC Byte 0 above).

Table 121: WoL MAC Byte 5, Index: 0x15 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL MAC Byte 5	RWR	Wake on LAN MAC Address Byte 5 (bit 7:0). Used as the WoL address to match n number of times (see description of WoL MAC Byte 0 above).

Table 122: WoL Password Byte 0, Index: 0x16 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL Pass Byte 0	RWR	Wake on LAN Password Byte 0 (bits 47:40). Used as the WoL password to match which follows immediately after the n number of WoL Address (if enabled in the port's WoL Control registers (index 0x1C and 0x1D of Switch MAC/WoL/WoF).

Table 123: WoL Password Byte 1, Index: 0x17 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL Pass Byte 1	RWR	Wake on LAN Password Byte 1 (bit 39:32). Used as the WoL password to match which follows immediately after the n number of WoL Address (see description of WoL MAC Byte 0 above).

Table 124: WoL Password Byte 2, Index: 0x18 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL Pass Byte 2	RWR	Wake on LAN Password Byte 2 (bit 31:24). Used as the WoL password to match which follows immediately after the n number of WoL Address (see description of WoL MAC Byte 0 above).

Table 125: WoL Password Byte 3, Index: 0x19 of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL Pass Byte 3	RWR	Wake on LAN Password Byte 3 (bit 23:16). Used as the WoL password to match which follows immediately after the n number of WoL Address (see description of WoL MAC Byte 0 above).

Table 126: WoL Password Byte 4, Index: 0x1A of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL Pass Byte 4	RWR	Wake on LAN Password Byte 4 (bit 15:8). Used as the WoL password to match which follows immediately after the n number of WoL Address (see description of WoL MAC Byte 0 above).

Table 127: WoL Password Byte 5, Index: 0x1B of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:0	WoL Pass Byte 5	RWR	Wake on LAN Password Byte 5 (bit 7:0). Used as the WoL password to match which follows immediately after the n number of WoL Address (see description of WoL MAC Byte 0 above).

Table 128: WoL Control Byte 0, Index: 0x1C of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:6	WoL Mode P3	RWR	Wake on LAN Mode Port 3. See WoL Mode P0 below.
5:4	WoL Mode P2	RWR	Wake on LAN Mode Port 2. See WoL Mode P0 below.
3:2	WoL Mode P1	RWR	Wake on LAN Mode Port 1. See WoL Mode P0 below.
1:0	WoL Mode P0	RWR	Wake on LAN Mode Port 0. The supported WoL Modes are: 00 = WoL disabled 01 = WoL enabled checking for 8 instances of the WoL Address 10 = WoL enabled checking for 16 instances of the WoL Address 11 = WoL enabled checking for 16 instances of the WoL Address followed by the WoL Password. When a WoL mode is enabled, and the selected number of WoL Addresses and Password are seen in a frame, the WoL Port register (index 0x1F) will be set to this port's number which in turn causes a Wake Event interrupt to be generated (Global 2 offset 0x00).

Table 129: WoL Control Byte 1, Index: 0x1D of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7:6	Reserved	RES	Reserved for future use.
5:4	WoL Mode P6	RWR	Wake on LAN Mode Port 6. See WoL Mode P4 below.
3:2	WoL Mode P5	RWR	Wake on LAN Mode Port 5. See WoL Mode P4 below.
1:0	WoL Mode P4	RWR	Wake on LAN Mode Port 4. The supported WoL Modes are: 00 = WoL disabled 01 = WoL enabled checking for 8 instances of the WoL Address 10 = WoL enabled checking for 16 instances of the WoL Address 11 = WoL enabled checking for 16 instances of the WoL Address followed by the WoL Password. When a WoL mode is enabled, and the selected number of WoL Addresses and Password are seen in a frame, the WoL Port register (index 0x1F) will be set to this port's number which in turn causes a Wake Event interrupt to be generated (Global 2 offset 0x00).

Table 130: WoL Control Byte 2, Index: 0x1E of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7	WoL MC MAC	RWR	Wake on LAN Multicast MAC address. When this bit is cleared to a zero the WoL MAC Address used to match is a unicast MAC address (i.e., WoL MAC Address bit 40 = 0). When this bit is set to a one the WoL MAC Address used to match is a multicast MAC address (i.e., WoL MAC Address bit 40 = 1). The rest of the WoL MAC Address bits are set in index 0x10 to 0x15 of Switch MAC/WoL/WoF above.
6:0	Reserved	RES	Reserved for future use.

Table 131: Wake Interrupt Source, Index: 0x1F of Switch MAC/WoL/WoF – G2 offset 0x0D

Bits	Field	Type	Description
7	WoF Int	RO	Wake on Frame Interrupt. This bit is set to a one if any of the WoF Int P[6:0] bits are set to a one in index 0x0C of Switch MAC/WoL/WoF above. When this bit is a one the Wake Event interrupt is set to a one (Global 2 offset 0x00). To clear out the Wake Event interrupt software need to examine the WoF Interrupt registers (index 0x0C) and clear the appropriate WoF Int En (Wake on Frame Interrupt Enable) bits (index 0x0E).
6:4	Reserved	RES	Reserved for future use.
3:0	WoL Port	RWS to 0xF	Wake on LAN Port. When a port is enabled with the WoL function (indexes 0x1C and 0x1D above) and a frame enters the port that matches the WoL criteria, this register will capture the port number where this event occurred. The changing of this register to a value other than 0xF will cause the Wake Event interrupt is set to a one (Global 2 offset 0x00). To clear out the Wake Event interrupt (and to to re-arm the WoL Port so that it can capture the port number of the next WoL event) software needs to write this register back to 0xF. **NOTE:** If a WoL event occurs while this register is holding a non-0xF value, that event will not cause a separate WoL interrupt (but the WoL frame will still be processed by the switch).

Table 132: ATU Stats
Offset: 0x0E or decimal 14 – G2

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 11:0 below. A value of 0x0 will access the 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 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: 0x0 = Count all valid entries 0x1 = Count all valid non-static entries only 0x2 = Count all valid entries found in the defined FID only 0x3 = 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 133: Priority Override Table
Offset: 0x0F or decimal 15 – G2

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 Qpri/Qpri_AVB or Fpri Override register selected by the FpriSet and Pointer bits below. After the write has taken place this bit self clears to zero.
14	FpriSet	RWR	Frame Priority Set of entries. When this bit is cleared to a zero the reading and writing actions of bits 7:0 below access the Qpri and Qpri_AVB entries of the Priority Override table for the frame type determined by the Pointer bits below. When this bit is set to a one, the reading and writing actions of bits 7:0 below access the Fpri entries of the Priority Override table for the frame type determined by the Pointer bits below. **NOTE:** This Priority Override table is accessed one set at a time (Qpri or Fpri) by this register interface.
13:12	Reserved	RES	Reserved for future use

Table 133: Priority Override Table (Continued)
Offset: 0x0F or decimal 15 – G2

Bits	Field	Type	Description
11:8	Pointer	RWR	Pointer to the desired entry of the Priority Override table. These bits select one of sixteen possible Qpri, Qpri_AVB or Fpri Override registers for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs. Each entry in the table can be used on the following frame types: 0x0 = Used on multicast DSA To_CPU frames with a Code of 0x0 (BPDU/MGMT) and on non-DSA multicast MGMT Control[1] 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) & 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) & 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) & 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) & 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) and on non-DSA unicast MGMT Control[2] 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. 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.

Table 133: Priority Override Table (Continued)
Offset: 0x0F or decimal 15 – G2

Bits	Field	Type	Description
11:8 (cont.)	Pointer	RWR	0xE = Used on Non-DSA Control frames that contain an Ethertype that matches 0x8863 (i.e., PPPoE frames). Not used on DSA Control frames. 0xF = Used on Non-DSA Control frames that contain an Ethertype that matches 0x0800 with a VER = 0x4 or an Ethertype that matches 0x86DD with a VER = 0x6 (i.e., Ipv4 or Ipv6 frames). Not used on DSA Control frames.
7	QpriAvbEn Only valid for devices that support AVB/TSN	RWR	AVB Queue Priority Override. When this entry's bit is set to a one the DataAvb bits below are used to override the frame's Qpri used for AVB enabled ports (AVB Port offset 0x0) or AvbOverride enabled ports (AVB Port offset 0x0). If this bit is cleared to a zero no QpriAvb override will occur for this entry. Up to sixteen QpriAvb override entries are possible in the table. What each entry is used for is defined in the Pointer bits above. Care is needed when setting this bit to a one. When a QpriAvb entry is enabled it will be applied to all frames of the entry's Type. The value in the entry's DataAvb bits will be the Qpri used when mapping all of these frames into an AVB enabled port as this data becomes the final QpriAvb for these frames. **NOTE:** This bit is accessible only when the FpriSet bit above equals zero.
6	Reserved	RES	Reserved for future use.
5:4	DataAvb Only valid for devices that support AVB/TSN	RWR	Queue Priority Override Data to use on AVB ports. 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[3] if it could be multiple Types) and the frame's Type is used to access this table. If the Type's QpriAvbEn bit (bit 7 above) is set to a one then the frame's QpriAvb will be overridden with the value found in this Data field[4]. **NOTE:** These bits are accessible only when the FpriSet bit above equals zero.
3	QpriEn Or FpriEn	RWR Except for entry 0x0 & 0x7 in the Qpri Set	Priority Override Enable. When FpriSet (bit above) equals zero and when this entry's bit is set to a one the lower two Data bits below are used to override the frame's Qpri used for non-AVB ports. If this bit is cleared to a zero no Qpri override will occur for this entry. When FpriSet (bit above) equals one and when this entry's bit is set to a one the Data bits below are used to override the frame's Fpri[5]. If this bit is cleared to a zero no Fpri override will occur for this entry. Up to sixteen Qpri and Fpri override entries are possible in the table. What each entry is used for is defined in the Pointer bits above.

Table 133: Priority Override Table (Continued)
Offset: 0x0F or decimal 15 – G2

Bits	Field	Type	Description
2:0	Data	RWR Except for entries 0x0 & 0x7 which are set to 0x3 in the Qpri Set.	Priority Override Data. When FpriSet (bit above) equals zero the lower two bits of this field (bits 1:0) are the entry's Queue Priority Override data (in this case the upper bit, bit 2, can be written as any value and the bit is undefined on reads). 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 FpriSet (bit above) equals one all three bits of this field are the entry's Frame Priority Override data. When a frame enters a port its Type is determined (in priority order[6] 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 fie[7]ld. If the Type's FpriEn bit (bit 3 above) is set to a one then the frame's Fpri will be overridden with the value found in this Data field.

1. Non-DSA multicast MGMT are multicast frames (not including broadcast frames) determined to be MGMT by a MGMT Entry State in the ATU (Global 1 offsets 0x0A to 0xF) or by the Rsvd2CPU mechanism (Global 2 offset 0x05).
2. Non-DSA unicast MGMT are unicast frames determined to be MGMT by a MGMT Entry State in the ATU (Global 1 offsets 0x0A to 0xF)
3. Priority order (low to high): Broadcast, PolMirror, PolTrap, ETYPE, PPPoE, IP, ARP, IGMP/MLD, MGMT.
4. 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).
5. If a frame's FPri is overridden by this table to be one of the two AVB FPri's (AvbHiFPri or AvbLoFPri in AVB Policy Global offset 0x00) the frame may be considered an AVB frame depending upon the ingress port's AvbMode (AVB Policy Port offset 0x00). In this case the frame's FPri may be modified if it egresses an AVB enabled port.
6. Priority order (low to high): Broadcast, PolMirror, PolTrap, ETYPE, PPPoE, IP, ARP, IGMP/MLD, MGMT.
7. If a frame can map to more that one item (like an ARP can also be a Broadcast) the last one in the priority order 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).

Table 134: I²C Block register, Global 2
Offset: 0x10 or decimal 16 – G2

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 8:0 will be loaded into the I²C Block register referenced by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:9	Pointer	RWR	Pointer to the desired 9-bit register of I²C Block. These bits select one of the possible I²C Block registers for both read and write operations. A write operation occurs if the Update bit is a one (the registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the desired register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 = I²C Transaction Control 0x01 = I²C Clock Control 0x02 to 0x0B = Reserved for future use 0x0C to 0xF = I²C Action Time 0x10 to 1F = I²C Addr/Data 16 entry Table for a single transaction 0x20 to 0x7F = Reserved for future use
8:0	Data	RWR	9-bit Data of the I²C Block register referenced by the Pointer bits above.

The individual registers accessed by the I²C Block register are described below and are indicated with a light blue background heading color.

Table 135: I2C Bus Transaction Control, Index: 0x00 of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Reserved	RES	Reserved for future use.
7	I²C Busy	SC	I²C Block Busy. 0 = I²C Block is idle 1 = I²C Block is busy with a transaction To start an I²C transaction: Set the I²C Speed, I²C Clock Mode & AckTime (index 0x01). Define the transaction's bytes in the I²C Addr/Data table (indexes 0x10 to 0x1F). Up to 16 byte transactions are supported. Optionally define the time the transaction should occur in the I²C Action Time registers (indexes 0x0C to 0x0F). Set ByteCnt to the size of the transaction, its I²C PTP mode (if appropriate) and set this bit to a one (which can all be done with the same write). When the I²C transaction completes this bit will self clear to zero. This bit transitioning from a one to a zero will set the I²C Done bit in Global 2 offset 0x13 index 0x03 bit 5. This bit being a one will set the EEInt bit in Global 1 offset 0x00 which in turn will drive the device's INTn pin low if it is unmasked.
6	I²C PTP	RWR	Time Release the I²C Transaction. 0 = I²C transaction will start as soon as the I²C Busy bit is set to a one 1 = I²C transaction will start based on PTP time When this bit is cleared to a zero, the I²C transaction will start as soon as the I²C Busy bit below is set to a one. When this bit is set to a one, the I²C transaction will start as soon as the I²C Busy bit below is set to a one and when the I²C Action Time (indexes 0x0C to 0x0F below) equal the PTP Global Time (TAI offset 0x0E & 0x0F). This supports time aware I²C transactions where they will start at a specific time.
5	AckTime Out	RO	Acknowledge Time Out occurred. 0 = Successful I²C Transaction if I²C Busy = zero 1 = Ack Timed Out if I²C Busy = 0 This bit is cleared whenever I²C Busy is set to a one. When I²C Busy returns to a zero this bit reflects the successful or unsuccessful completion of the I²C transaction.
4	Reserved	RES	Reserved for future use.
3:0	ByteCnt	RWR	I²C Transaction Byte Count. These bits are used to define the size of the I²C transaction in bytes. Use a value of 0x1 for 1 byte, a value of 0x2 for 2 bytes, etc. Use a value of 0x0 for 16 byte transaction.

Table 136: I2C Bus Clock Control, Index: 0x01 of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Reserved	RES	Reserved for future use.
7	Abort	RWR	I²C Abort. 0 = Normal operation 1 = Reset the I²C block and interface When this bit is set to a one, the I²C block and interface will be reset. The current transaction, if any, will be aborted wherever it is – even if it is in the middle of a transaction byte. If it was in the middle of a transaction the ByteCnt will indicate the byte # where this abort occurred. This block 'reset' state will be held until this bit is cleared to zero. **NOTE:** Only the I²C block is reset by this bit and this bit will not affect any writable register values.
6	I²C Clock Mode	RWR	I²C Clock Mode. 0 = The I²C Clock toggles only during an I²C transaction 1 = The I²C Clock toggles continuously When this bit is set to a one the I²C Clock will toggle continuously until this bit is cleared to a zero. When this bit is cleared to zero the I²C Clock will toggle only during I²C transactions.
5:4	I²C Speed	RWR	I²C Transaction clock Speed. The following I²C clock speeds are supported: 0x0 = 100 kHz 0x1 = 400 kHz 0x2 = 1700 kHz 0x3 = 3400 kHz The hardware needs to get as close to these clock rates without going over. This must be implemented with a glitch free MUX.

Table 136: I2C Bus Clock Control, Index: 0x01 of I²C Block – G2 Offset 0x10 (Continued)

Bits	Field	Type	Description
3:0	AckTime	RWR	I²C Maximum Ack Time. This register is used to determine how long the I²C interface is to wait for an Acknowledge from the addressed I²C Slave device. If the I²C Slave does not Ack within this prescribed time the I²C Transaction will terminate (I²C Busy in index 0x00 is cleared to zero) and the AckTimeOut bit will be set to a one (index 0x00). The AckTIme is defined as the number of I²C clocks after the last bit of the byte of data is transmitted to the slave. A value of 0x0 disables the AckTImeOut feature such that the I²C interface will wait forever for an Ack. In this case the Abort bit above can be used to get back control of a stalled interface. **NOTE:** This feature only counts I²C Clocks that are not pulled low by the slave.

Table 137: I2C Action Time byte 0, Index: 0x0C of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Reserved	RES	Reserved for future use.
7:0	ActionTime [7:0]	RWR	I²C Transaction Time bits 7:0 of a 32-bit value. When the I²C Busy bit is set to a one along with the I²C PTP bit being a one (both in index 0x00), the I²C transaction will start only when the PTP Global Time (TAI offsets 0x0E & 0x0F) equals the value in the 32-bit Action Time register (this field being bits 7:0 of that register).

Table 138: I2C Action Time byte 1, Index: 0x0D of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Reserved	RES	Reserved for future use.
7:0	ActionTime [15:8]	RWR	I²C Transaction Time bits 15:8 of a 32-bit value. See the description for ActionTime[7:0] above.

Table 139: I2C Action Time byte 2, Index: 0x0E of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Reserved	RES	Reserved for future use.

Table 139: I2C Action Time byte 2, Index: 0x0E of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
7:0	ActionTime [23:16]	RWR	I²C Transaction Time bits 23:16 of a 32-bit value. See the description for ActionTime[7:0] above.

Table 140: I2C Action Time byte 3, Index: 0x0F of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Reserved	RES	Reserved for future use.
7:0	ActionTime [31:24]	RWR	I²C Transaction Time bits 32:24 of a 32-bit value. See the description for ActionTime[7:0] above.

Table 141: I2C Bus Addr/Data, Index: 0x10 to 0x1F of I²C Block – G2 Offset 0x10

Bits	Field	Type	Description
8	Start	RWR	Start Byte Indicator. 0 = The bits in the Addr/Data register below are 8-bits of Addr/Data 1 = The bits in the Addr/Data register below are 7-bits of Addr & 1 R/W bit The I²C transaction begins by processing the 1st entry in this table. After its 8-bits are sent out and the acknowledgement is received from the slave, the ByteCnt (index 0x00) is decremented. If the ByteCnt is now zero the I²C transaction ends. If the ByteCnt is non-zero the next entry in this table is processed. The I²C transaction must begin with a Start byte which contains 7 address bits and 1 R/W bit. The 1st entry in this table is this Start byte and thus the Start bit must be set to a one in the 1st entry in this table. The 2nd byte in the I²C transaction typically reads or writes 8-bits of data or an additional 8-bits of address. Any byte (like these that don't contain a R/W bit) must clear the Start bit to zero for its entry in the table. Any subsequent bytes in the I²C transaction that transfer 8-bits of data (either in or out) must clear this bit to zero in that byte's entry in the table. Some I²C transactions require one or more subsequent Start bytes in the middle of the transaction. These ReStart bytes have the same format as the initial Start byte (7-bits of Addr & 1 R/W bit) and these bytes in the table must have this Start bit set to a one on it entry in the table. The hardware will issue an I²C Start or ReStart clock cycle whenever it processes a table entry where this Start bit is a one. It will issue an I²C Stop clock cycle when it decrements the ByteCnt to zero.

Table 141: I2C Bus Addr/Data, Index: 0x10 to 0x1F of I^{2}C Block – G2 Offset 0x10 (Continued)

Bits	Field	Type	Description
7:0	Addr/Data	RWR	Address Data Field. This field has two possible formats depending upon the Start bit above: If the Start bit = 0: This field contains 8-bits of Addr or Data If the Start bit = 1: Bits 7:1 of this field is Addr & Bit 0 is R/W When the entry's Start bit is cleared to zero, the 8-bits in this field are: • The lower 8-bits of a 10-bit address, or • They are 8-bits of data to write (it the last R/W bit was zero), or • They are the storage space of 8-bits of data to read (if the last R/W bit was one) When the entry's Start bit is set to a one, the 8-bits in this field are: • 7 address bits (in bits 7:1 of this register) & the R/W bit (in bit 0) Set the R/W bit to 1 for Read operations and clear it to 0 for Write operations. Set the 7 address bits to the address of the device or use 0b11110xx to use a 10-bit address where xx are the upper 2 address bits.

Note: The device contains sixteen I2C Bus Addr/Data entries in the above table. This supports I^{2}C transactions that are up to 16 bytes in length.

Table 142: Energy Management Register
Offset: 0x12 or decimal 18 – G2

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 Energy Management register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	Pointer to the desired octet of Energy Management. These bits select one of the possible Energy Management registers for both read and write operations. A write operation occurs if the Update bit is a one (the Energy Management registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the Energy Management register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 to 0x07 = Reserved for future use 0x08 to 0x0F = Energy Efficient Ethernet (EEE) register space 0x10 to 0x7F = Reserved for future use All other addresses are reserved for future use.
7:0	Data	RWR	Energy Management data read or written to the register pointed to by the Pointer bits above.

The individual registers accessed by the Energy Management register are described below and are indicated with a light yellow background heading color.

Table 143: EEE Timer Rates, Index: 0x08 of Energy Management – G2 Offset 0x12

Bits	Field	Type	Description
7:6	Reserved	RES	Reserved for future use.
5:4	AsrtRate	RWR	Assertion Timer Rate. These bits determine the resolution of the Assertion Timer at register index 0x0A as follows: 0x0 = 1 uSec steps (default) 0x1 = 1 mSec steps 0x2 = 1 Sec steps 0x3 = Reserved
3:2	WakeRate	RWR	Wake Timer Rate. These bits determine the resolution of the Wake Timers at register indexes 0x0B (for FE ports) and 0x0C (for GE ports) as follows: 0x0 = 1 uSec steps (default) 0x1 = 1 mSec steps 0x2 = 1 Sec steps 0x3 = Reserved
1:0	TxIdleRate	RWS to 0x1	TxIdle Timer Rate. These bits determine the resolution of the Tx Idle Timer at register index 0x0F as follows: 0x0 = 1 uSec steps 0x1 = 1 mSec steps (default) 0x2 = 1 Sec steps 0x3 = Reserved

Table 144: EEE Assertion Timer, Index: 0x0A of Energy Management – G2 Offset 0x12

Bits	Field	Type	Description
7:0	AsrtTime	RWR	EEE Assertion Time. When the PHY is asked to enter low power mode it cannot immediately leave this mode. This timer determines the time requirement for the PHY to fully power down before a wake up request can be processed. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by AsrtRate in Index 0x08 above. An AsrtTime of 0x0 has the low power idle code deasserted immediately when an idle request is deasserted.

Table 145: EEE Wake Timer FE, Index: 0x0B of Energy Management – G2 Offset 0x12

Bits	Field	Type	Description
7:0	WakeTime FE	RWS to 0x1E	EEE Wake Time Fast Ethernet ports. In order for PHY to exit low power mode and return back to normal mode, a wake up period is needed. These bits determine the time that switch must wait for the PHY to completely exit low power mode before the switch can start to transmit a packet. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by WakeRate in Index 0x08 above. The default value of 0x1E is 30uSec.

Table 146: EEE Wake Timer GE, Index: 0x0C of Energy Management – G2 Offset 0x12

Bits	Field	Type	Description
7:0	WakeTime GE	RWS to 0x11	EEE Wake Time for Gigabit Ethernet ports. In order for PHY to exit low power mode and return back to normal mode, a wake up period is needed. These bits determine the time that switch must wait for the PHY to completely exit low power mode before the switch can start to transmit a packet. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by WakeRate in Index 0x08 above. The default value of 0x11 is 17uSec.

Table 147: EEE TxIdle Timer, Index: 0x0F of Energy Management – G2 Offset 0x12

Bits	Field	Type	Description
7:0	TxIdleTime	RWS to 0x02	EEE Transmit Idle Time. These bits determine the time that each TX port must remain idle before telling the PHY to entering its low power state. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by TxIdleRate in Index 0x08 above. If the Transmit Idle Time is set to 0x0 then automatic low power mode is entered as soon as the port's egress queue is empty and has nothing more to send.

Table 148: IMP Communication/Debug Register
Offset: 0x13 or decimal 19 – G2

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 IMP Communications/Debug register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWS to 0x03[1]	Pointer to the desired octet of IMP Communications/Debug. These bits select one of the possible IMP Communications/Debug registers for both read and write operations. A write operation occurs if the Update bit is a one (the IMP Communications/Debug registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the IMP Communications/Debug register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 to 0x04 = Communication Interface to/from the CPU The following registers are also supported, but are not documented here. Instead these registers are documented in the IMP Functional Spec for this device: 0x05 to 0x07 = Reserved for future use 0x08 to 0x0F = Debug Control & Status 0x10 to 0x1F = CPU's Main and Alternate Registers (always readable) 0x20 to 0x29 = CPU's Special Purpose Registers (always readable) 0x2A to 0x2E = Reserved for future use 0x2F = IMP I/O Page Register 0x30 to 0x33 = Break Point Control 0x38 = Break Point Status 0x39 to 0x3F = Reserved for future use 0x40 to 0x47 = Break Point 0 0x48 to 0x4F = Break Point 1 0x50 to 0x57 = Break Point 2 0x58 to 0x5F = Break Point 3 0x60 to 0x6F = Reserved for future use 0x70 to 0x7F = IMP I/O Registers (always readable)
7:0	Data	RWR	IMP Communications/Debug data read or written to the register pointed to by the Pointer bits above.

1. The Pointer resets to 0x03 so that the sources from the EEInt (Global 1 offset 0x00) can be accessed quickly.

The individual registers accessed by the IMP Communications & Debug register are described below and are indicated with a light green background heading color.

Table 149: Comm Status, Index: 0x00 of IMP Comm/Debug – G2 Offset 0x13

Bits	Field	Type	Description
7	CPU Stopped	RO	CPU clocks are Stopped. 0 = CPU is running 1 = CPU clocks are stopped This bit is set to a one whenever the IMP CPU's clocks are stopped. The clocks can be stopped by the IMP CPU writing to its I/O register 0x02 (by setting the Total IMP Stop or the CPUStop bits to a one), by executing a Halt instruction, or by never starting in the first place (i.e., no EEPROM is present or the EEPROM was not tagged for the IMP CPU's use).
6:5	WtFlags	RWR	Write Data Flags. These two bits can be used for any purpose of communicating extra information between this interface and the IMP. For example, one of these bits can be used to indicate if the WrData contains a Command or Data.
4	WtDataBsy	RO	Write Data to the IMP is Busy. 0 = The WtData register is available for new information 1 = The WtData register is busy with data for the IMP This bit is automatically set to a one whenever the WtData register (Index 0x02) is written to by this interface. It automatically clears to a zero whenever the IMP reads this data out of this register. This bit being a one can be used to interrupt the IMP.
3	RdIntEn	RWR	Enable the RdData Interrupt from the IMP. 0 = The RdData Interrupt from the IMP is not active 1 = The RdData Interrupt from the IMP is active Setting this bit to a one enables the IMP to drive this device's interrupt pin through the EEInt interface (Global 1 offset 0x00 & Index 0x03 below). The device's interrupt pin will go low (if EEInt is enabled in Global 1 offset 0x04) when this bit is set to a one and whenever the RdDataRdy bit below is also a one. When the RdData register (Index 0x01) is read by this interface this interrupt will be cleared until the next write to the RdData register by the IMP.
2:1	RdFlags	RO	Read Data Flags. These two bits can be used for any purpose of communicating extra information between the IMP and this interface. For example, one of these bits can be used to indicate if the RdData contains a Command or Data.

Table 149: Comm Status, Index: 0x00 of IMP Comm/Debug – G2 Offset 0x13 (Continued)

Bits	Field	Type	Description
0	RdDataRdy	RO	Data from the IMP is ready to be Read. 0 = Data in the RdData register is not valid 1 = Data in the RdData register is valid and should be read This bit being set to a one indicates that the IMP has placed a byte of data in the RdData register (Index 0x01). When the RdData register is read by this interface, this bit will automatically clear to zero. This bit being a one can be used to drive this device's interrupt pin if the RdIntEn bit above is set to a one.

Table 150: Comm Read Data, Index: 0x01 of IMP Comm/Debug – G2 Offset 0x13

Bits	Field	Type	Description
7:0	RdData	RO	Read Data from the IMP. This register contains the byte of data the IMP has sent to this interface. It contains valid data whenever the RdDataRdy bit (Index 0x00) is set to a one. When this RdData register is read, the RdDataRdy bit will be cleared to a zero indicating to the IMP that this register is ready for the next byte of data. The IMP can cause an EEInt interrupt to occur whenever valid data from the IMP is contained in this register. See Index 0x00.

Table 151: Comm Write Data, Index: 0x02 of IMP Comm/Debug – G2 Offset 0x13

Bits	Field	Type	Description
7:0	WrData	RWR	Write Data to the IMP. This register contains the byte of data this interface wants to send to the IMP. Writing to this register automatically sets the WtDataBsy bit which will not be cleared until the IMP reads this register's contents. Therefore, this register should not be written to again without first checking that the WtDataBsy bit is zero. If a write is done with the WtDataBsy bit being a one, the write will occur, overwriting the contents of this register. This interface can cause an interrupt to occur to the IMP CPU whenever valid data to the IMP is contained in this register. See RdIntEn at IMP I/O address 0x04.

Table 152: EE Interrupt Source, Index: 0x03 of IMP Comm/Debug – G2 Offset 0x13

Bits	Field	Type	Description
7:6	Reserved	RES	Reserved for future use.
5	I^2C Done	ROC	I^2C operation Done. This bit gets set to a one whenever the I2C Busy bit (Global 2 offset 0x10 index 0x00) transitions from a one to a zero. This bit self clears when read.
4	EEPROM Done	ROC	EEPROM operation Done. This bit gets set to a one whenever the EEBusy bit (Global 2 offset 0x14) transitions from a one to a zero. This bit self clears when read.
3	Break	ROC	IMP Break. This bit is set to a one whenever a Break occurs for a Break Point that has its BkXIntEn bit set to a one (where X is the number of the Break Point – at Indexes 0x30 to 0x33). This bit self clears when read. Which Break Point is active (and therefore the one that caused this bit to be set to a one) can be seen by reading Index 0x38.
2	RdRdy	RO	IMP Read Data Ready. This bit is set to a one whenever the IMP's RdDataRdy bit is a one and the RdIntEn bit is a one (both bits are in Index 0x00 above).
1	RLDone	ROC	Register Loader Done. This bit is set to a one whenever the Register Loader was given control by the EEPROM and the Register Loader has executed a HALT instruction. This bit self clears when read.
0	NoEEP	ROC	No EEPROM. This bit is set to a one whenever the first byte read from the EEPROM was 0xFF, meaning no EEPROM was present. This bit self clears when read.

Note: If any of these bits in this register are a one the EEInt in Global 1 offset 0x00 will be set to a one, which in turn will drive the device's INTn pin low if the EEIntEn in Global 1 offset 0x04 is also set to a one.

Table 153: IMP State, Index: 0x04 of IMP Comm/Debug – G2 Offset 0x13

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	EEPROM Read	RO	EEPROM Read in process. This bit is set to a one whenever the EEPROM is performing a read operation. This bit will be cleared to zero once the EEPROM read completes or if the EEPROM read timed out. If an EEPROM read time out occurred on other than the 1st EEPROM read, the timed out EEPROM read will be re-tried (and this bit will be set to a one again).
2	EEPROM TimedOut	RO	EEPROM Timed Out. This bit is set to a one whenever any EEPROM read operation timed out. This bit will be cleared to a zero whenever a Reset IMPOp occurs (Global 2 offset 0x13 index 0x08).
1	Total IMP Stop	RO	Total IMP Stop. This bit is a copy of the Total IMP Stop bit at IMP I/O address 0x02 except this bit is read only. This bit will be cleared to a zero whenever a Reset IMPOp occurs (Global 2 offset 0x13 index 0x08).
0	No EEPROM	RO	No EEPROM. This bit is a copy of the NoEEP bit in index 0x03, except that this bit does not clear on read. This bit will be cleared to a zero whenever a Reset IMPOp occurs (Global 2 offset 0x13 index 0x08).

Table 154: EEPROM Command
Offset: 0x14 or decimal 20 – G2

Bits	Field	Type	Description
15	EEBusy	SC	EEPROM Unit Busy. This bit must be set to a one to start an EEPROM operation (see EEOp below). Only one EEPROM operation can be executing at one time so this bit must be zero before setting it to a one. When the requested EEPROM 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 (the EEInt in Global 1, offset 0x00).
14:12	EEOp	RWR	EEPROM Opcode. The device supports the following EEPROM operations (all of these operations can be executed while frames are transiting through the switch): 0x0 = No Operation 0x1 = Reserved 0x2 = Reserved 0x3 = Write byte to EEPROM at Addr (Addr from EEPROM Addr register below – Global 2 offset 0x15) 0x4 = Read byte from EEPROM from Addr (Addr from EEPROM Addr register below – Global 2 offset 0x15) 0x5 = Reserved 0x6 = Restart Register Loader execution at Addr 0x7 = Reserved
11	Running	RO	Register Loader Running. This bit is set to a one whenever the register loader is busy executing the instructions contained in the EEPROM. An EEPROM Read or Write can only be done when this bit is zero.
10	WriteEn	RWR	EEPROM Write Enable. This bit must be set to a one before writing to the EEPROM is possible. If this bit is a zero the Write EEPROM EEOp above will not do anything.
9:8	Reserved	RES	Reserved for future use
7:0	Data	RWR	Data to/from the EEPROM. Data to be written to the EEPROM or data that was read back from the EEPROM. The EEPROM action (read or write) and the location of the action (the address inside the EEPROM) are defined in the EEPROM Command register above.

Table 155: EEPROM Addr
Offset: 0x15 or decimal 21 – G2

Bits	Field	Type	Description
15:0	Addr	RWR	EEPROM Address. This is the EEPROM's address where the EEOp (above) is performed and whenever the EEPROM stops executing it contains the address of the last EEPROM command executed.

Table 156: AVB Command Register
Offset: 0x16 – G2

Bits	Field	Type	Description
15	AVBBusy	SC	AVB unit Busy. This bit must be set to a one to start an AVB operation (see AVBOp below). Only one AVB operation can be executing at one time so this bit must be zero before setting it to a one. When the requested AVB operation completes, this bit will automatically be cleared to a zero.
14:13	AVBOp Only valid for devices that support AVB/TSN	RWR	AVB unit Operation code. The devices support the following AVB operations (all of these operations can be executed while frames are transiting through the switch): 0x0 = Read from the register pointed to by AVBAddr below. The data read from the selected register is transferred to AVBData register. 0x1 = Reserved 0x2 = Read with post increment of register address defined in bits AVBAddr bits below. For PTP data structures, this command instructs the hardware to take a snap-shot of 4 consecutive data registersstarting with the AVBAddr location. This is used for capturing time counter values which are more than 16 bits wide along with the sequence identifier information. 0x3 = Write to the register pointed by AVBAddr below. AVBData registerscontent gets written into the selected register.
12:8	AVBPort	RWR	This indicates the physical port of this device. These bits indicate the AVB port that is being accessed in the AVBCommand register. For example, if this field is programmed to a 0x1, it indicates that AVB registers belonging to port number 1 are being accessed. To access PTP global registers, set the AVBBlock to 0x0 and set AVBPort to 0x1F. To access Time Application Interface (TAI) Global registers, set the AVBBlock to 0x0 and set AVBPort to 0x1E. To access AVB Policy global registers, set the AVBBlock to 0x1 and set AVBPort to 0x1F. To access Qav global registers, set the AVBBlock to 0x2 and set AVBPort to 0x1F. To access Qbv global registers, set the AVBBlock to 0x2 and set AVBPort to 0x1F (but there are no Qbv global registers at this time).

Table 156: AVB Command Register
Offset: 0x16 – G2

Bits	Field	Type	Description
7:5	AVBBlock	RWR	AVB Block. This field indicates the block of addresses within the device. For example within the AVB register space, this feld selects the block like PTP, SRP and Qav etc. The AVBAddr field below selects the specific address within a selected block. 0x0 = To select PTP register space (documented in section X p X) 0x1 = To select AVB Policy register space (documented in section X p X) 0x2 = To select Qav register space (documented in section X p X) 0x3 = To select Qbv register space (documented in section X p X) 0x4 – 0x7 = Reserved for future use. Note that accessing registers in the reserved range of the AVBBlock would return all zero's back for the AVBCommand register.
4:0	AVBAddr	RWR	AVB Address. These bits indicate the address bits for the register operation being specified in the AVBOp bits specified above.

Table 157: AVB Data Register
Offset: 0x17 or decimal 23 – G2

Bits	Field	Type	Description
15:0	AVBData Only valid for devices that support AVB/TSN	RWR	AVB Data bits. These data bits indicate either the read data or the write data bits depending on the AVB Command register (Global 2 offset 0x16). In the case of a read operation, the hardware logic fetches the data bits from the specified address in AVB Command register and stores them into these bits. In the case of a write operation, the hardware logic utilizes the data bits in this field to write to the specified address location in AVB Command register. The content of the AVB registers is documented by AVBBlock (see AVBBlock bits in the AVB Command register above).

Table 158: SMI PHY Command for SMI Access
Offset: 0x18 or decimal 24 – G2

Bits	Field	Type	Description
15	SMIBusy	SC	SMI PHY Unit Busy. This bit must be set to a one to start an SMI operation. 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.
14:13	SMIFunc	RWR	SMI Function. These bits are used to define what bits[12:0] of this register do as follows: 0x0 = Internal Access - The register bits as described below (along with the SMI PHY Data Register – Global 2 offset 0x19) are used to access the internal PHYs and/or SERDES contained inside this device. In this mode the external MDC_PHY & MDIO_PHY pins will not toggle. 0x1 = External Access - The register bits as described below (along with the SMI PHY Data Register – Global 2 offset 0x19) are used to access external devices connected to the device's MDC_PHY & MDIO_PHY pins. In this mode the internal PHY SMI bus will not toggle allowing all 32 DevAddrs to be accessed. 0x2 = SMI Setup - See the description of this register below labeled for SMI Setup 0x3 = 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): 0x0 = Reserved 0x1 = Write Data Register 0x2 = Read Data Register 0x3 = Reserved When the SMIMode bit = 0 then SMIOp = (IEEE 802.3 Clause 45): 0x0 = Write Address Register 0x1 = Write Data Register 0x2 = Read Data Register with post increment on the Address Register 0x3 = Read Data Register

Table 158: SMI PHY Command for SMI Access (Continued)
Offset: 0x18 or decimal 24 – G2

Bits	Field	Type	Description
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.

Table 159: SMI PHY Data Register
Offset: 0x19 or decimal 25 – G2

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 PHY operation (i.e., before setting SMIBusy to a one). During SMI PHY Reads these bits will contain the SMI 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.

Table 160: SMI PHY Command Register for SMI Setup
Offset: 0x18 or decimal 24 – G2

Bits	Field	Type	Description
15	SMIBusy	SC	SMI PHY Unit Busy. This bit must be set to a one to start an SMI operation. 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	SMIFunc	RWR	SMI Function. These bits are used to define what bits[12:0] of this register do as follows: 0x0 = SMI Access - See the description of this register above labeled for SMI Access 0x1 = SMI Access - See the description of this register above labeled for SMI Access 0x2 = SMI Setup - The register bits as described below (along with the SMI PHY Data Register – Global 2 offset 0x19) are used to access the SMI Registers. 0x3 = Reserved for future use.
12	Reserved	RES	Reserved for future use.
11:10	SMIOp	RWR	SMI PHY Opcode. These bits are used to select the SMI opcode to operate on during SMI commands as follows: 0x0 = Reserved 0x1 = Write Data Register 0x2 = Read Data Register 0x3 = Reserved
9:0	Pointer	RWR	Pointer to the desired register of SMI Setup. These bits select one of the possible SMI Setup registers for both read and write operations. A write operation occurs if the SMIOp is a Write. Otherwise a read of the current Pointer occurs and the data found there is placed in the SMI Data register bits below (Global 2 offset 0x19). The Pointer bits are used to access the Index registers as follows: 0x000 to 0x01F = SMI Device Address Translation Table. The data stored at the Pointer's location is the SMI DevAddr the PPU will use when accessing the external PHY connected to that physical port number. The reset state of this table is <entry's data> = Pointer (i.e., there is no translation by default). External accesses using the SMI External Access method (above) are not affected by this table. 0x020 to 0x3FF = Reserved for future use.

Table 161: SMI PHY Data Register
Offset: 0x19 or decimal 25 – G2

Bits	Field	Type	Description
15:0	SMIData	RWR	SMI Data register. During SMI Setup Writes these bits must be written with the SMI data to be written prior to starting the SMI write operation (i.e., before setting SMIBusy to a one). During SMI Setup Reads these bits will contain the SMI data that was read after the SMI read operation completes (i.e., SMIBusy returns to a zero). Writes to this register must not be done while SMIBusy is a one. Bits 15:4 must be zero when writing to the SMI Device Address Translation Table.

Table 162: Scratch and Misc
Offset: 0x1A or decimal 26 – G2

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 Scratch and Misc. Control register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	Pointer to the desired octet of Scratch and Misc (Miscellaneous). These bits select one of the possible Scratch and Misc registers for both read and write operations. A write operation occurs if the Update bit is a one (the Scratch and Misc registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the Scratch and Misc register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 to 0x01 = Scratch Bytes 0x02 = Misc Configuration 0x03 to 0x1F = Reserved for future use 0x20 to 0x3F = GPIO Port Stall Vectors 0x60 to 0x6F = GPIO registers data and configuration 0x70 to 0x7F = CONFIG reads All other addresses are reserved for future use.
7:0	Data	RWR	Scratch and Misc Data. Control data read or written to the register pointed to by the Pointer bits above.

The individual registers accessed by the Scratch and Misc. Control register are described below and are indicated with a light blue background heading color.

Table 163 shows which GPIO pins are available depending upon the mode of Port 0's interface pins.

Table 163: GPIO Availability Configuration Matrix

Port 0's C_Mode	PHY Detect	Name	0	1	2	3	4	5	6	7	8	9	10
0x0	X	FD MII	x	x	x	x	x	x	x	√	√	√	√
0x1	X	MII PHY	x	x	x	x	x	x	x	x	x	√	√
0x2	0	MII MAC	x	x	x	x	x	x	x	x	x	√	√
	1	MII to PHY	x	x	x	x	x	x	x	x	x	√	√
0x3	X	GMII	-	-	-	-	-	-	-	-	-	-	-
0x4	0	RMII PHY	x	x	x	x	x	√	√	√	√	√	√
	1	RMII to PHY	x	x	x	x	x	√	√	√	√	√	√
0x5	0	RMII MAC	x	x	x	x	x	√	√	√	√	√	√
	1	RMII to PHY	x	x	x	x	x	√	√	√	√	√	√
0x6	X	xMII Tristate	√	√	√	√	√	√	√	√	√	√	√
0x7	0	RGMII	x	x	x	x	x	x	x	√	√	√	√
	1	RGMII to PHY	x	x	x	x	x	x	x	√	√	√	√

Table 164: Scratch Byte 0, Index: 0x00 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	Scratch Byte 0	RWR	Scratch bits. These bits are 100% available to software for whatever purpose desired. These bits do not connect to any hardware function. NOTE: These bits are cleared to zero with a hardware reset.

Table 165: Scratch Byte 1, Index: 0x01 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	Scratch Byte 1	RWR	Scratch bits. These bits are 100% available to software for whatever purpose desired. These bits do not connect to any hardware function. NOTE: These bits are cleared to zero with a hardware reset.

Table 166: Misc Configuration, Index: 0x02 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	NormalSMI	RES but resets to 0x1	Normal SMI vs. GPIO mode. When P0_MODE is not equal to 0x1 or 0x2 the P0_COL and P5_CRS pins are not needed. In this case, when this bit is set to a one, the P0_COL and P0_CRS pins become MDIO_PHY and MDC_PHY, respectively, if the NO_CPU configuration pin was a one during reset. Else the pins become GPIO pins. Clearing this bit to a zero inverts the effect the NO_CPU configuration pin's value has on the function of the P0_COL and P0_CRS pins when they are not needed (i.e., when P0_MODE is not equal to 0x1 or 0x2).
6:0	Reserved	RES	Reserved for future use.

Table 167: GPIO 0 Port Stall Vector 0, Index: 0x20 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 0 Port Stall Vector [6:0]	RWR	GPIO 0's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 0's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 0's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 0 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 0 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 168: GPIO 0 Port Stall Vector 1, Index: 0x21 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 0 Port Stall En	RWR	GPIO 0's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 0 pin, if the pin is currently a GPIO input, is used to activate GPIO 0's Port Stall Vector (Index 0x20 above). When this bit is cleared to a zero GPIO 0's Port Stall Vector function is disabled.
6	GPIO 0 Port Stall Value	RWR	GPIO 0's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 0 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 0 Port Stall En bit (above) is set to a one, the GPIO 0 Port Stall Vector (Index 0x20 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 0 Port Stall Vector – Index 0x20 above).
5:0	Reserved	RES	Reserved for future use.

Table 169: GPIO 1 Port Stall Vector 0, Index: 0x22 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 1 Port Stall Vector [6:0]	RWR	GPIO 1's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 1's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 1's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 1 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 1 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 170: GPIO 1 Port Stall Vector 1, Index: 0x23 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 1 Port Stall En	RWR	GPIO 1's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 1 pin, if the pin is currently a GPIO input, is used to activate GPIO 1's Port Stall Vector (Index 0x22 above). When this bit is cleared to a zero GPIO 1's Port Stall Vector function is disabled.
6	GPIO 1 Port Stall Value	RWR	GPIO 1's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 1 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 1 Port Stall En bit (above) is set to a one, the GPIO 1 Port Stall Vector (Index 0x22 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 1 Port Stall Vector – Index 0x22 above).
5:0	Reserved	RES	Reserved for future use.

Table 171: GPIO 2 Port Stall Vector 0, Index: 0x24 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 2 Port Stall Vector [6:0]	RWR	GPIO 2's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 2's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 2's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 2 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 2 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 172: GPIO 2 Port Stall Vector 1, Index: 0x25 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 2 Port Stall En	RWR	GPIO 2's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 2 pin, if the pin is currently a GPIO input, is used to activate GPIO 2's Port Stall Vector (Index 0x24 above). When this bit is cleared to a zero GPIO 2's Port Stall Vector function is disabled.
6	GPIO 2 Port Stall Value	RWR	GPIO 2's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 2 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 2 Port Stall En bit (above) is set to a one, the GPIO 2 Port Stall Vector (Index 0x24 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 2 Port Stall Vector – Index 0x24 above).
5:0	Reserved	RES	Reserved for future use.

Table 173: GPIO 3 Port Stall Vector 0, Index: 0x26 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 3 Port Stall Vector [6:0]	RWR	GPIO 3's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 3's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 3's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 3 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 3 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 174: GPIO 3 Port Stall Vector 1, Index: 0x27 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 3 Port Stall En	RWR	GPIO 3's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 3 pin, if the pin is currently a GPIO input, is used to activate GPIO 3's Port Stall Vector (Index 0x26 above). When this bit is cleared to a zero GPIO 3's Port Stall Vector function is disabled.
6	GPIO 3 Port Stall Value	RWR	GPIO 3's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 3 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 3 Port Stall En bit (above) is set to a one, the GPIO 3 Port Stall Vector (Index 0x26 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 3 Port Stall Vector – Index 0x26 above).
5:0	Reserved	RES	Reserved for future use.

Table 175: GPIO 4 Port Stall Vector 0, Index: 0x28 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 4 Port Stall Vector [6:0]	RWR	GPIO 4's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 4's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 4's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 4 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 4 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 176: GPIO 4 Port Stall Vector 1, Index: 0x29 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 4 Port Stall En	RWR	GPIO 4's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 4 pin, if the pin is currently a GPIO input, is used to activate GPIO 4's Port Stall Vector (Index 0x28 above). When this bit is cleared to a zero GPIO 4's Port Stall Vector function is disabled.
6	GPIO 4 Port Stall Value	RWR	GPIO 4's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 4 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 4 Port Stall En bit (above) is set to a one, the GPIO 4 Port Stall Vector (Index 0x28 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 4 Port Stall Vector – Index 0x28 above).
5:0	Reserved	RES	Reserved for future use.

Table 177: GPIO 5 Port Stall Vector 0, Index: 0x2A of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 5 Port Stall Vector [6:0]	RWR	GPIO 5's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 5's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 5's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 5 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 5 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 178: GPIO 5 Port Stall Vector 1, Index: 0x2B of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 5 Port Stall En	RWR	GPIO 5's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 5 pin, if the pin is currently a GPIO input, is used to activate GPIO 5's Port Stall Vector (Index 0x2A above). When this bit is cleared to a zero GPIO 5's Port Stall Vector function is disabled.
6	GPIO 5 Port Stall Value	RWR	GPIO 5's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 5 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 5 Port Stall En bit (above) is set to a one, the GPIO 5 Port Stall Vector (Index 0x2A above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 5 Port Stall Vector – Index 0x2A above).
5:0	Reserved	RES	Reserved for future use.

Table 179: GPIO 6 Port Stall Vector 0, Index: 0x2C of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 6 Port Stall Vector [6:0]	RWR	GPIO 6's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 6's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 6's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 6 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 6 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 180: GPIO 6 Port Stall Vector 1, Index: 0x2D of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 6 Port Stall En	RWR	GPIO 6's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 6 pin, if the pin is currently a GPIO input, is used to activate GPIO 6's Port Stall Vector (Index 0x2C above). When this bit is cleared to a zero GPIO 6's Port Stall Vector function is disabled.
6	GPIO 6 Port Stall Value	RWR	GPIO 6's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 6 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 6 Port Stall En bit (above) is set to a one, the GPIO 6 Port Stall Vector (Index 0x2C above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 6 Port Stall Vector – Index 0x2C above).
5:0	Reserved	RES	Reserved for future use.

Table 181: GPIO 7 Port Stall Vector 0, Index: 0x2E of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 7 Port Stall Vector [6:0]	RWR	GPIO 7's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 7's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 7's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 7 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 7 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 182: GPIO 7 Port Stall Vector 1, Index: 0x2F of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 7 Port Stall En	RWR	GPIO 7's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 7 pin, if the pin is currently a GPIO input, is used to activate GPIO 7's Port Stall Vector (Index 0x2E above). When this bit is cleared to a zero GPIO 7's Port Stall Vector function is disabled.
6	GPIO 7 Port Stall Value	RWR	GPIO 7's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 7 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 7 Port Stall En bit (above) is set to a one, the GPIO 7 Port Stall Vector (Index 0x2E above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 7 Port Stall Vector – Index 0x2E above).
5:0	Reserved	RES	Reserved for future use.

Table 183: GPIO 8 Port Stall Vector 0, Index: 0x30 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 8 Port Stall Vector [6:0]	RWR	GPIO 8's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 8's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 8's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 8 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 8 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 184: GPIO 8 Port Stall Vector 1, Index: 0x31 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 8 Port Stall En	RWR	GPIO 8's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 8 pin, if the pin is currently a GPIO input, is used to activate GPIO 8's Port Stall Vector (Index 0x30 above). When this bit is cleared to a zero GPIO 8's Port Stall Vector function is disabled.
6	GPIO 8 Port Stall Value	RWR	GPIO 8's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 8 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 8 Port Stall En bit (above) is set to a one, the GPIO 8 Port Stall Vector (Index 0x30 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 8 Port Stall Vector – Index 0x30 above).
5:0	Reserved	RES	Reserved for future use.

Table 185: GPIO 9 Port Stall Vector 0, Index: 0x32 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 9 Port Stall Vector [6:0]	RWR	GPIO 9's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 9's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 9's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 9 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 9 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

GPIO 9 Port Stall Vector 1, Index: 0x33 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 9 Port Stall En	RWR	GPIO 9's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 9 pin, if the pin is currently a GPIO input, is used to activate GPIO 9's Port Stall Vector (Index 0x32 above). When this bit is cleared to a zero GPIO 9's Port Stall Vector function is disabled.
6	GPIO 9 Port Stall Value	RWR	GPIO 9's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 9 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 9 Port Stall En bit (above) is set to a one, the GPIO 9 Port Stall Vector (Index 0x32 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 9 Port Stall Vector – Index 0x32 above).
5:0	Reserved	RES	Reserved for future use.

Table 186: GPIO 10 Port Stall Vector 0, Index: 0x34 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 10 Port Stall Vector [6:0]	RWR	GPIO 10's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 10's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 10's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 10 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 10 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is de-asserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 187: GPIO 10 Port Stall Vector 1, Index: 0x35 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO 10 Port Stall En	RWR	GPIO 10's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 10 pin, if the pin is currently a GPIO input, is used to activate GPIO 10's Port Stall Vector (Index 0x34 above). When this bit is cleared to a zero GPIO 10's Port Stall Vector function is disabled.
6	GPIO 10 Port Stall Value	RWR	GPIO 10's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 10 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 10 Port Stall En bit (above) is set to a one, the GPIO 10 Port Stall Vector (Index 0x34 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 10 Port Stall Vector – Index 0x34 above).
5:0	Reserved	RES	Reserved for future use.

Table 188: GPIO Configuration, Index: 0x60 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	GPIO[7] Mode	RO	General Purpose Input Output [7]'s Mode. GPIO[7] is shared with P0_COL. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_COL pin is not needed and if the NO_CPU configuration pin is a zero[1], this pin becomes GPIO[7] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[7], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 3 (Register Index 0x6B of Scratch and Misc. Control).
6	GPIO[6] Mode	RO	General Purpose Input Output [6]'s Mode. GPIO[6] is shared with P0_IND[3]. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_IND[3] pin is not needed, this pin becomes GPIO[6] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[6], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 3 (Register Index 0x6B of Scratch and Misc. Control).
5	GPIO[5] Mode	RO	General Purpose Input Output [5]'s Mode. GPIO[5] is shared with P0_IND[2]. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_IND[2] pin is not needed, this pin becomes GPIO[5] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[5], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 2 (Register Index 0x6A of Scratch and Misc. Control).

Table 188: GPIO Configuration, Index: 0x60 of Scratch/Misc. – G2 Offset 0x1A (Continued)

Bits	Field	Type	Description
4	GPIO[4] Mode	RO	General Purpose Input Output [4]'s Mode. GPIO[4] is shared with P0_IND[1]. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_IND[1] pin is not needed, this pin becomes GPIO[4] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[4], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 2 (Register Index 0x6A of Scratch and Misc. Control).
3	GPIO[3] Mode	RO	General Purpose Input Output [3]'s Mode. GPIO[3] is shared with P0_IND[0]. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_IND[0] pin is not needed, this pin becomes GPIO[3] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[3], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 1 (Register Index 0x69 of Scratch and Misc. Control).
2	GPIO[2] Mode	RO	General Purpose Input Output [2]'s Mode. GPIO[2] is shared with P0_INDV. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_INDV pin is not needed, this pin becomes GPIO[2] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[2], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 1 (Register Index 0x69 of Scratch and Misc. Control).

Table 188: GPIO Configuration, Index: 0x60 of Scratch/Misc. – G2 Offset 0x1A (Continued)

Bits	Field	Type	Description
1	GPIO[1] Mode	RO	General Purpose Input Output [1]'s Mode. GPIO[1] is shared with P0_INCLK. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_INCLK pin is not needed, this pin becomes GPIO[1] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[1], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 0 (Register Index 0x68 of Scratch and Misc. Control).
0	GPIO[0] Mode	RO	General Purpose Input Output [0]'s Mode. GPIO[0] is shared with P0_OUTCLK. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_OUTCLK pin is not needed, this pin becomes GPIO[0] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[0], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 0 (Register Index 0x68 of Scratch and Misc. Control).

1. The effect of the NO_CPU configuration pin's value can be inverted. See NormalSMI at index 0x02 of this register.

Table 189: GPIO Configuration, Index: 0x61 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:3	Reserved	RES	Reserved for future use.
2	GPIO[10] Mode	RO	General Purpose Input Output [10]'s Mode. GPIO[10] is a dedicated pin so this bit will always be a one. When configured to be GPIO[10], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 5 (Register Index 0x6D of Scratch and Misc. Control).
1	GPIO[9] Mode	RO	General Purpose Input Output [9]'s Mode. GPIO[9] is a dedicated pin so this bit will always be a one. When configured to be GPIO[9], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 4 (Register Index 0x6C of Scratch and Misc. Control).
0	GPIO[8] Mode	RO	General Purpose Input Output [8]'s Mode. GPIO[8] is shared with P0_CRS. When Port 0 is configured into a mode (by the P0_MODE pins) where the P0_CRS pin is not needed and if the NO_CPU configuration pin is a zero[1], this pin becomes GPIO[8] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[8], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or written is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 4 (Register Index 0x6C of Scratch and Misc. Control).

1. The effect of the NO_CPU configuration pin's value can be inverted. See NormalSMI at index 0x02 of this register.

Table 190: GPIO Direction, Index: 0x62 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	GPIO Direction [7:0]	RWS to 0xFF	General Purpose Input Output's Direction. This register is used to control the direction of GPIO[7:0]. Bit 0 controls GPIO[0], bit 1 controls GPIO[1], etc. When a GPIO's bit is set to a one that GPIO will become an input. When a GPIO's bit is cleared to a zero that GPIO will become an output. This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x60) above.

GPIO Direction, Index: 0x63 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:3	Reserved	RES	Reserved for future use.
2:0	GPIO Direction [10:8]	RWS to 0x07	General Purpose Input Output's Direction. This register is used to control the direction of GPIO[10:8]. Bit 0 controls GPIO[8], bit 1 controls GPIO[9], etc. When a GPIO's bit is set to a one that GPIO will become an input. When a GPIO's bit is cleared to a zero that GPIO will become an output. This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x61) above.

Table 191: GPIO Data, Index: 0x64 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	GPIO Data [7:0]	RWR	General Purpose Input Output's Data. This register is used to access the data of GPIO[7:0]. Bit 0 accesses GPIO[0], bit 1 accesses GPIO[1], etc. When a GPIO's bit is set to be an input (by the GPIO Direction bits above, Index 0x62) data written to this bit will go to a holding register but will not appear on the pin nor in this register. Reads of this register will return the actual, real-time, data that is appearing on the GPIO's pin. When a GPIO's bit is set to be an output (by the GPIO Direction bits above, Index 0x62) data written to this bit will go to a holding register and will appear on the GPIO's pin. Reads of this register will return the actual, real-time, data that is appearing on the GPIO's pin (which in this case should be the data written, but if its isn't that would be an indication of a conflict). When a pin's direction changes from input to output, the data last written to the holding register appears on the GPIO's pin. This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x60). Note: Any GPIO that is not enabled (i.e., its Mode bit is zero in Index 0x60) will return a zero on reads.

Table 192: GPIO Data, Index: 0x65 of Scratch & Misc. – Offset 0x1A

Bits	Field	Type	Description
7:0	GPIO Data [15:8]	RWR	General Purpose Input Output's Data. This register is used to access the data of GPIO[15:8]. Bit 0 accesses GPIO[8], bit 1 accesses GPIO[9], etc. When a GPIO's bit is set to be an input (by the GPIO Direction bits above, Index 0x63) data written to this bit will go to a holding register but will not appear on the pin nor in this register. Reads of this register will return the actual, real-time, data that is appearing on the GPIO's pin. When a GPIO's bit is set to be an output (by the GPIO Direction bits above, Index 0x63) data written to this bit will go to a holding register and will appear on the GPIO's pin. Reads of this register will return the actual, real-time, data that is appearing on the GPIO's pin (which in this case should be the data written, but if its isn't that would be an indication of a conflict). When a pin's direction changes from input to output, the data last written to the holding register appears on the GPIO's pin. This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x61). Note: Any GPIO that is not enabled (i.e., its Mode bit is zero in Index 0x61) will return a zero on reads. This includes GPIO bits that don't exist in this device, i.e., GPIO Data [15:11].

Table 193: GPIO Pin Control 0, Index: 0x68 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[1] Control	RWR	General Purpose Input Output Interface pin 1 Control. This register is used to control alternate functions of the GPIO[1] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[1] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[1] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use
3	Reserved	RES	Reserved for future use.
2:0	GPIO[0] Control	RWR	General Purpose Input Output Interface pin 0 Control. This register is used to control alternate functions of the GPIO[0] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[0] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[0] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use

Note: If more than one GPIO pin is configured for the exact same function, the lowest number GPIO pin will receive the function and the higher numbered GPIO pins will retain the GPIO function. These register bits will not reflect this situation, however, but the GPIO Configuration register (Register index 0x60 of Scratch and Misc. Control) will reflect that the higher numbered GPIO pin is still a GPIO pin (unless its an MII pin).

Table 194: GPIO Pin Control 1, Index: 0x69 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[3] Control	RWR	General Purpose Input Output Interface pin 3 Control. This register is used to control alternate functions of the GPIO[3] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[3] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[3] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use
3	Reserved	RES	Reserved for future use.
2:0	GPIO[2] Control	RWR	General Purpose Input Output Interface pin 2 Control. This register is used to control alternate functions of the GPIO[2] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[2] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[2] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use

Note: If more than one GPIO pin is configured for the exact same function, the lowest number GPIO pin will receive the function and the higher numbered GPIO pins will retain the GPIO function. These register bits will not reflect this situation, however, but the GPIO Configuration register (Register index 0x60 of Scratch and Misc. Control) will reflect that the higher numbered GPIO pin is still a GPIO pin (unless its an MII pin).

Table 195: GPIO Pin Control 2, Index: 0x6A of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[5] Control	RWR	General Purpose Input Output Interface pin 5 Control. This register is used to control alternate functions of the GPIO[5] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[5] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[5] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use
3	Reserved	RES	Reserved for future use.
2:0	GPIO[4] Control	RWR	General Purpose Input Output Interface pin 4 Control. This register is used to control alternate functions of the GPIO[4] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[4] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[4] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use

Note: If more than one GPIO pin is configured for the exact same function, the lowest number GPIO pin will receive the function and the higher numbered GPIO pins will retain the GPIO function. These register bits will not reflect this situation, however, but the GPIO Configuration register (Register index 0x60 of Scratch and Misc. Control) will reflect that the higher numbered GPIO pin is still a GPIO pin (unless its an MII pin).

Table 196: GPIO Pin Control 3, Index: 0x6B of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[7] Control	RWR	General Purpose Input Output Interface pin 7 Control. This register is used to control alternate functions of the GPIO[7] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[7] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[7] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = I2C_SDA – I^2C data pin if GPIO[8] is I2C_SCL
3	Reserved	RES	Reserved for future use.
2:0	GPIO[6] Control	RWR	General Purpose Input Output Interface pin 6 Control. This register is used to control alternate functions of the GPIO[6] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[6] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[6] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = Reserved for future use

Note: If more than one GPIO pin is configured for the exact same function, the lowest number GPIO pin will receive the function and the higher numbered GPIO pins will retain the GPIO function. These register bits will not reflect this situation, however, but the GPIO Configuration register (Register index 0x60 of Scratch and Misc. Control) will reflect that the higher numbered GPIO pin is still a GPIO pin (unless its an MII pin).

Table 197: GPIO Pin Control 4, Index: 0x6C of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[9] Control	RWR	General Purpose Input Output Interface pin 9 Control. This register is used to control alternate functions of the GPIO[9] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[9] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[9] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = I2C_SDA – I^2C data pin if GPIO[10] is I2C_SCL
3	Reserved	RES	Reserved for future use.
2:0	GPIO[8] Control	RWR	General Purpose Input Output Interface pin 8 Control. This register is used to control alternate functions of the GPIO[8] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[8] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[8] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = I2C_SCL – I^2C clock pin for I2C_SDA on GPIO[7]

Note: If more than one GPIO pin is configured for the exact same function, the lowest number GPIO pin will receive the function and the higher numbered GPIO pins will retain the GPIO function. These register bits will not reflect this situation, however, but the GPIO Configuration registers (Register index 0x60 & 0x61 of Scratch and Misc. Control) will reflect that the higher numbered GPIO pin is still a GPIO pin (unless its an MII pin).

Table 198: GPIO Pin Control 5, Index: 0x6D of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:3	Reserved	RES	Reserved for future use.
2:0	GPIO[10] Control	RWR	General Purpose Input Output Interface pin 10 Control. This register is used to control alternate functions of the GPIO[10] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[10] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 0x0 = GPIO[10] 0x1 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 0x2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 0x3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 0x4 = RX_CLK0/Qbv0 – SyncE Receive Clock or Qbv Window 0 Output 0x5 = RX_CLK1/Qbv1 – SyncE Receive Clock or Qbv Window 1 Output 0x6 = Reserved for future use 0x7 = I2C_SCL – I^2C clock pin for I2C_SDA on GPIO[9]

Note: If more than one GPIO pin is configured for the exact same function, the lowest number GPIO pin will receive the function and the higher numbered GPIO pins will retain the GPIO function. These register bits will not reflect this situation, however, but the GPIO Configuration register (Register index 0x60 & 0x61 of Scratch and Misc. Control) will reflect that the higher numbered GPIO pin is still a GPIO pin (unless its an MII pin).

Table 199: CONFIG Data0, Index: 0x70 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	Config 0	RO	Reset Configuration pin data 0. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values is registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 0's bits are: Bit CONFIG Pin's Primary Name 0 Reserved - 1 Reserved - 2 Reserved - 3 ADDR[0] R0_LED 4 ADDR[1] R1_LED 5 ADDR[2] R2_LED 6 ADDR]3] R3_LED 7 ADDR[4] No Pin – Always reads as zero **NOTE:** The Address CONFIG pins are called ADDR[4:0]n so the value in this register is inverted from what was 'seen' on the pins on the rising edge of RESETn.

Table 200: CONFIG Data1, Index: 0x71 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	Config 1	RO	Reset Configuration pin data 1. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values is registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 1's bits are: Bit CONFIG Pin's Primary Name 0 LED_SEL C2_LED 1 Reserved - 2 Reserved - 3 Reserved - 4 Reserved - 5 Reserved - 6 FLOW EE_CLK/C0_LED 7 Reserved -

Table 201: CONFIG Data2, Index: 0x72 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	Config 2	RO	Reset Configuration pin data 2. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values is registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 1's bits are: Bit CONFIG Pin's Primary Name 0 P0_MODE[0] P0_OUTD[0] 1 P0_MODE[1] P0_OUTD[1] 2 P0_MODE[2] P0_OUTD[2] 3 Reserved - 4 Reserved - 5 Reserved - 6 Reserved - 7 Reserved -

Table 202: CONFIG Data3, Index: 0x73 of Scratch/Misc. – G2 Offset 0x1A

Bits	Field	Type	Description
7:0	Config 3	RO	Reset Configuration pin data 3. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values is registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 1's bits are: Bit CONFIG Pin's Primary Name 0 P5_SMODE C3_LED 1 Reserved - 2 Reserved - 3 Reserved - 4 P0_VDDOS P0_OUTEN 5 NO_CPU P0_OUTD[3] 6 Reserved - 7 Reserved -

Table 203: Watch Dog Control Register
Offset: 0x1B or decimal 27 – G2

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 Watch Dog Control register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	Pointer to the desired octet of Watch Dog. These bits select one of the possible Watch Dog registers for both read and write operations. A write operation occurs if the Update bit is a one (the Watch Dog registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the Watch Dog register can be read by first writing to this register, with Update = 0, and then reading this register). The Pointer bits are used to access the Index registers as follows: 0x00 = Watch Dog Interrupt Source 0x10 to 0x13 = Data Path Watch Dog Interrupts, Masks, Events & History 0x40 = Auto Fixing Enables All other addresses are reserved for future use.
7:0	Data	RWR	Watch Dog Control data read or written to the register pointed to by the Pointer bits above.

The individual registers accessed by the Watch Dog Control register are described below and are indicated with a light purple background heading color.

Table 204: Watch Dog Interrupt Source, Index: 0x00 of Watch Dog – G2 offset 0x1B

Bits	Field	Type	Description
7:1	Reserved	RES	Reserved for future use.
0	DataPath WDInt	RO	Data Path Watch Dog Interrupt. This bit is set to a one whenever any of the Data Path Watch Dog Events is set to a one in Watch Dog index 0x10. This bit being a one will cause the WatchDogInt bit to be a one (Global 2 index 0x00).

Table 205: Data Path Watch Dog Interrupts, Index: 0x10 of Watch Dog – G2 offset 0x1B

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD Int	RO	Cut Through Watch Dog Interrupt. This bit will be set to a one whenever the CT Watch Dog Func (index 0x11) is set to a one and the CT Watch Dog Event (index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (index 0x00).
2	QC WD Int	RO	Queue Controller Watch Dog Interrupt. This bit will be set to a one whenever the QC Watch Dog Func (index 0x11) is set to a one and the QC Watch Dog Event (index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (index 0x00).
1	EgressWD Int	RO	Egress Watch Dog Interrupt. This bit will be set to a one whenever the Egress Watch Dog Func (index 0x11) is set to a one and the Egress Watch Dog Event (index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (index 0x00).
0	ForceWD Int	RO	Force Watch Dog Interrupt. This bit will be set to a one whenever the Force Watch Dog Func (index 0x11) is set to a one and the Force Watch Dog Event (index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (index 0x00).

Table 206: Data Path Watch Dog Masks, Index: 0x11 of Watch Dog – G2 offset 0x1B

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD Func	RWS	Cut Through Watch Dog Function. 0 = CT WD generates a SWReset (if enabled – index 0x13) 1 = CT WD generates a CT WD Int (index 0x10) When this bit is set to a one Cut Through Watch Dog Events (index 0x12) will generate Cut Through Watch Dog interrupts (index 0x10). This will in turn cause a Watch Dog interrupt (Global 2 offset 0x00) which will drive the Interrupt pin if Watch Dog interrupts are unmasked (Global 2 offset 0x01). When this bit is cleared to a zero Cut Through Watch Dog Events (index 0x12) will cause a Switch Software Reset (Global 1 offset 0x04) if the SWReset on WD bit is set to a one (index 0x13).
2	QC WD Func	RWS	Queue Controller Watch Dog Function. 0 = QC WD generates a SWReset (if enabled – index 0x13) 1 = QC WD generates a CT WD Int (index 0x10) When this bit is set to a one Queue Controller Watch Dog Events (index 0x12) will generate Queue Controller Watch Dog interrupts (index 0x10). This will in turn cause a Watch Dog interrupt (Global 2 offset 0x00) which will drive the Interrupt pin if Watch Dog interrupts are unmasked (Global 2 offset 0x01). When this bit is cleared to a zero Queue Controller Watch Dog Events (index 0x12) will cause a Switch Software Reset (Global 1 offset 0x04) if the SWReset on WD bit is set to a one (index 0x13).
1	EgressWD Func	RWS	Egress Watch Dog Function. 0 = Egress WD generates a SWReset (if enabled – index 0x13) 1 = Egress WD generates a CT WD Int (index 0x10) When this bit is set to a one Egress Watch Dog Events (index 0x12) will generate Egress Watch Dog interrupts (index 0x10). This will in turn cause a Watch Dog interrupt (Global 2 offset 0x00) which will drive the Interrupt pin if Watch Dog interrupts are unmasked (Global 2 offset 0x01). When this bit is cleared to a zero Egress Watch Dog Events (index 0x12) will cause a Switch Software Reset (Global 1 offset 0x04) if the SWReset on WD bit is set to a one (index 0x13).

Table 206: Data Path Watch Dog Masks, Index: 0x11 of Watch Dog – G2 offset 0x1B (Continued)

Bits	Field	Type	Description
0	ForceWD Func	RWS	Force a Watch Dog Function. 0 = Force WD generates a SWReset (if enabled – index 0x13) 1 = Force WD generates a CT WD Int (index 0x10) When this bit is set to a one Forced Watch Dog Events (index 0x12) will generate Forced Watch Dog interrupts (index 0x10). This will in turn cause a Watch Dog interrupt (Global 2 offset 0x00) which will drive the Interrupt pin if Watch Dog interrupts are unmasked (Global 2 offset 0x01). When this bit is cleared to a zero Forced Watch Dog Events (index 0x12) will cause a Switch Software Reset (Global 1 offset 0x04) if the SWReset on WD bit is set to a one (index 0x13).

Table 207: Data Path Watch Dog Events, Index: 0x12 of Watch Dog – G2 offset 0x1B

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD Event	RO	Cut Through Watch Dog Event. If the Cut Through logic detects any CT watch dog issues, this bit will be set to a one, regardless of the setting of the CT Watch Dog Func bit (index 0x11). This bit is cleared by a SWReset (Global 1 offset 0x04). It will automatically be cleared to zero if the SWReset on WD bit (index 0x13) is set to a one and this event's Func bit (CT WD Func) is cleared to zero (index 0x11).
2	QC WD Event	RO	Queue Controller Watch Dog Event. If the QC logic detects any QC watch dog issues, this bit will be set to a one, regardless of the setting of the QC Watch Dog Func bit (index 0x11). This bit is cleared by a SWReset (Global 1 offset 0x04). It will automatically be cleared to zero if the SWReset on WD bit (index 0x13) is set to a one and this event's Func bit (QC WD Func) is cleared to zero (index 0x11).
1	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 Func bit (index 0x11). This bit is cleared by a SWReset (Global 1 offset 0x04). It will automatically be cleared to zero if the SWReset on WD bit (index 0x13) is set to a one and this event's Func bit (Egress WD Func) is cleared to zero (index 0x11).
0	ForceWD Event	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 SWReset (Global 1 offset 0x04). It will automatically be cleared to zero if the SWReset on WD bit (index 0x13) is set to a one and this event's Func bit (Force WD Func) is cleared to zero (index 0x11).

Table 208: Data Path Watch Dog History, Index: 0x13 of Watch Dog – G2 offset 0x1B

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD History	RO	Cut Through Watch Dog History. If the Cut Through logic detects any CT watch dog issues, this bit will be set to a one, regardless of the setting of the CT Watch Dog Func bit (index 0x11). This bit can only be cleared by a hardware reset.
2	QC WD History	RO	Queue Controller Watch Dog History. If the QC logic detects any QC watch dog issues, this bit will be set to a one, regardless of the setting of the QC Watch Dog Func bit (index 0x11). This bit can only be cleared by a hardware reset.
1	EgressWD 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 Func bit (index 0x11). 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, watch dog events (index 0x12) whose associated Func bit (index 0x11) is cleared to a zero will automatically reset the switch core's datapath just as if the SWReset bit (Global 1 offset 0x04) was set to a one. The Watch Dog History bits, above 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 watch dog events (index 0x12) will not cause a SWReset.

Table 209: Auto Fixing Enables, Index: 0x40 of Watch Dog – G2 offset 0x1B

Bits	Field	Type	Description
7	RMU TimeOut	RWS	Remote Management Timout. 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.
6:4	Reserved	RES	Reserved for future use.
3:0	DMPort	RWS to 0xF	Duplex Mismatch Port. When a Duplex Missmatch interrupt event occurs (Global 2 offset 0x00) these bits will reflect the port that generated the interrupt. Software needs to set these bits back to 0xF in order to re-arm the Duplex Mismatch interrupt event so that future Duplex Mismatch events can be reported.

Table 210: QoS Weights Register
Offset: 0x1C or decimal 28 – G2 offset 0x1B

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 211: Misc Register
Offset: 0x1D or decimal 29 – G2 offset 0x1B

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.
12:0	Reserved	RES	Reserved for future use.

1. Source Device is used if the frame's Src_Is_Trunk = 0, else this device's Device Number is used.

Table 212: Cut Through Control Register
Offset: 0x1F or decimal 31 – G2 offset 0x1B

Bits	Field	Type	Description
15:13	Reserved	RES	Reserved for future use.
12:8	Cut Through Hold	RWR	Cut Through Burst Hold amount. To support bursts of frames in Cut Through mode, once a Cut Through connection is made between ports, the Cut Through connection needs to be held beyond the end of each transmitted frame. This hold time keeps the last Cut Through connection active until the next ingressing frame can be processed to see if it is also to be Cut Through. If the Cut Through Hold time is too small the Cut Through fabric will disconnect before the next ingressing frame can be Cut Through allowing any Store and Forward frames that are queued up to egress the port. If the Cut Through Hold time is too long the port will be idle longer than it needs to be before switching to the Store and Forward queues and transmitting any frames that may be stored there. The Cut Through Hold register determines the number of octets a Cut Through connection is held after the last bytes of a frame's CRC is transmitted. The default value of 0x00 breaks the connection right away. A value of 0x16 (22 decimal) will hold the connection for a 96-bit IFG + 64-bit Preamble with 2 bytes of pad in case the IFG expanded due to PPM clock differences.
7:0	Reserved	RES	Reserved for future use.

1.3.4 Ingress Rate Limiter (IRL) Registers

The device contains a set of port Ingress Rate Limiting (IRL) registers that effect the selected Ethernet ports in the device. Each PIRL register is 16-bits wide and their bit assignments are shown in Table 213 on page 213.

There are five sets of these registers per port, one set per IRL 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).

Each of these five resources (selected by IRLRes numbers 0x0 to 0x4 – Global 2 offset 0x09) are optimized for two modes of operation as follows:

- Internet Protocol (IP) rate limiting and Broadcast Storm prevention types of functions. This mode also support a Ingress Mirroring Sampling mode and is discussed starting on page ???<<section 1.3.4.1>>.
- AVB stream rate policing types of functions, specifically for policing Qav flows. This mode is also called ITSM for Isochronous Time Slow Metering. These resources are discussed starting on page ???<<section 1.3.4.2>>.

The mode of operation a resource is in is controlled by the resource's TM bit (TSM Mode – offset 0x07). When this bit is a zero the IP/Storm mode of operation is chosen. When this bit is a one the AVB/ITSM mode of operation is chosen.

1.4 IRL IP/Storm Limiting Registers (IRLRes 0x0 to 0x4)

The IP/Storm Rate Limiting resources are discussed below. There are five (5) sets of these registers per port, i.e., one set per IRL 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) when the IRL Resource pointer (IRLRes) has a value of 0x0 to 0x4 and the resource's TM bit (offset 0x07) is cleared to a zero.

Table 213: IRL IP/Storm Limiting Resource Register bit Map (from Global 2 offsets 0x09 & 0x0A)

IP/Storm Bucket Configuration Register Data Bits

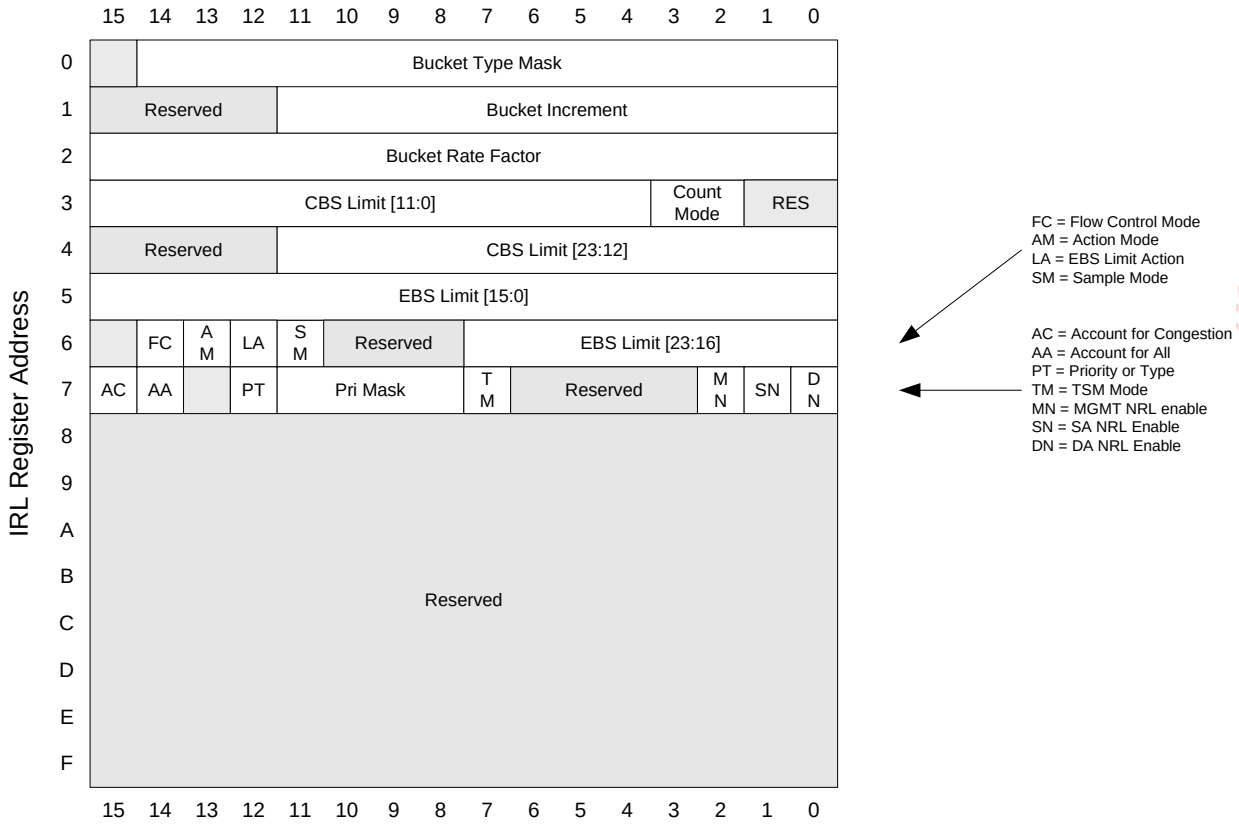


Table 214: IRL Bucket Configuration Register
Offset: 0x0 or decimal 0

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:0	BktType Mask	RWR	This field has the following definition if BktRateType = 0; [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] – Flow 0 or TCP Data [9] – Flow 1 or TCP Ctrl (if any of the TCP FLAGS[5:0] bits are set) [10] – Flow 2 or UDP [11] – Flow 3 or NON_TCPUDP (covers IGMP, ICMP, GRE, IGRP, L2TP) [12] – IMS (Ingress Monitor Source) [13] – PolicyMirror [14] – PolicyTrap When TSMMode is 0x1, this field is reserved. Flow [3:0] are selected for bits 11 to 8 of this register when TCAM Flows = 1 (IRL offset 0x07). One of the Flow [3:0] wires will be set to a one when the TCAM FlowID is valid for this frame[1] and FlowID[5] is zero. FlowID[1:0] determine which Flow [3:0] wire is set to a one, where Flow 0 will be a 1 if FlowID[1:0] = 0, Flow 1 will be a 1 if FlowID[1:0] = 1, etc. If FlowID[5] = 1 then none of the Flow [3:0] wires will be set to a one such that no PIRL matches on them can be made. **NOTE:** Flow[3:0] are only valid on devices that contain a TCAM.

1. The TCAM FlowID will be valid for a TCAM hit on a frame where the TCAM Continue Action bit is a zero. See the TCAM Action registers 1 and 2 in TCAM Page 2, offsets 0x02 and 0x03.

Table 215: IRL Bucket Configuration Register
Offset: 0x1 or decimal 1

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future uses.
11:0	BktInc	RWR	Bucket Increment This parameter indicates the amount of tokens that need to be added for each byte of frame information. When TSMMode is 0x1, this field is reserved.

Table 216: IRL 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. When TSMMode is 0x1, this field is reserved.

Table 217: IRL 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	Count Mode. 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: 0x0 = Frame based 0x1 = Count all Layer1 bytes 0x2 = Count all Layer2 bytes 0x3 = 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 218: IRL Bucket Configuration Register
Offset: 0x4 or decimal 4

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future uses.
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). When TSMMode is 0x1, the CBSLimit bits 23 down to 16 are reserved.

Table 219: IRL 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 220: IRL Bucket Configuration Register
Offset: 0x6 or decimal 6

Bits	Field	Type	Description
15	Reserved	RWR	Reserved for future use.
14	IrlFC Mode	RWR	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. When TSMMode is 0x1, this field is reserved.
13	Action Mode[1]	RWR	IRL 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	EBS Limit Action. 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 IRL resource buckets for this port were to discard the packet the packet would get discarded by the switch and similarly if any of the IRL resource buckets for this port were to send a flow control packet back to the source the flow control packets does get sent. When TSMMode is 0x1, this field is reserved.

Table 220: IRL Bucket Configuration Register (Continued)
Offset: 0x6 or decimal 6

Bits	Field	Type	Description
11	SMode	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 IMS or PolMirror bin in 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 device, 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. When TSMMode is 0x1, this field is reserved.
10:8	Reserved	RES	Reserved for future uses.
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 that if ActionMode=1, the BktTokenCount can exceed EBSLimit and if the ActionMode = 0, the BktTokenCount is clamped at EBSLimit. When TSMMode is 0x1, the EBSLimit bits 23 down to 16 are reserved.

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 221: IRL Bucket Configuration Register
Offset: 0x7 or decimal 7

Bits	Field	Type	Description
15	AcctFor QCong	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. When TSMMode is 0x1, this field is reserved.
14	AcctForAll	RWR	Account for all 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 policy reasons, or they are discarded due to errors or Port States, account for the incoming bytes (or frames if the CountMode is configured to be 0x0) anyway in the bucket_increment calculations. Basically this mode counts all the bytes of data that came 'down the wire' into the switch port. If SMode = 1, AcctForAll 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, or they are discarded due to errors or Port States, do not account for the incoming bytes (or frames if the CountMode is configured to be 0x0) in the bucket_increment calculations. Basically this mode counts the frames that were accepted into the switch memory. When TSMMode is 0x1, this field is reserved.
13	Reserved	RES	Reserved for future uses.

Table 221: IRL Bucket Configuration Register (Continued)
Offset: 0x7 or decimal 7

Bits	Field	Type	Description
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. When TSMMode is 0x1, this field is reserved.
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	TSM Mode	RWR	Time Slot Metering mode. This bit must be 0x0 for IP/Storm Rate Limiting. When 0x0, the rate bucket operates in the normal information rate limiting mode of operation. In this mode, the bucket operates in IP/Storm Rate Limiting mode and all the features described in this section are applicable.
6:3	Reserved	RES	Reserved for future use.
3	TCAM Flows Only valid on devices that contain a TCAM	RWR	TCAM Flows. Enable the TCAM for Flow identifiers for Flows 0 to 3. When this bit is set to a one, bits 11:8 of the BktTypeMask are connected to the TCAM's FlowID Action bits. In this way, up to four flows per port can be limited and/or counted. When this bit is a zero BktTypeMask bits 11:8 mean NON_TCPUDP, UDP, TCP Ctrl and TCP Data, respectively.

Table 221: IRL Bucket Configuration Register (Continued)
Offset: 0x7 or decimal 7

Bits	Field	Type	Description
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	SaAvbNrl En	RWR	Source Address AVB / Non Rate Limit enable. 0 = Account for all frames as enabled in this resource 1 = Do not account for SA non rate limiting overrides/SA AVB flows 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 AVB/Non Rate Limited[2]. When this occurs the frame will not be ingress rate limited. When TSMMode is 0x1, this field is reserved.
0	DaAvbNrl En	RWR	Destination Address AVB / Non Rate Limit enable. 0 = Account for all frames as enabled in this resource 1 = Do not account for DA non rate limiting overrides/DA AVB flows 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 AVB/Non Rate Limited. When this occurs the frame will not be ingress rate limited. When TSMMode is 0x1, this field is reserved.

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.

1.5 IRL ITSM Limiting Registers (IRLRes 0x0 to 0x4)

The AVB Stream Rate Limiting resources are discussed below. **These features are not valid in devices that do not support AVB.**

There are five (5) sets of these registers per port, i.e., one set per IRL 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) when the IRL Resource pointer (IRLRes) has a value of 0x0 to 0x04 and the resource's TM bit (offset 0x07) is set to a one.

Table 222: IRL ITSM Limiting Resource Register bit Map (from Global 2 offsets 0x09 & 0x0A)

ITSM Bucket Configuration Register Data Bits

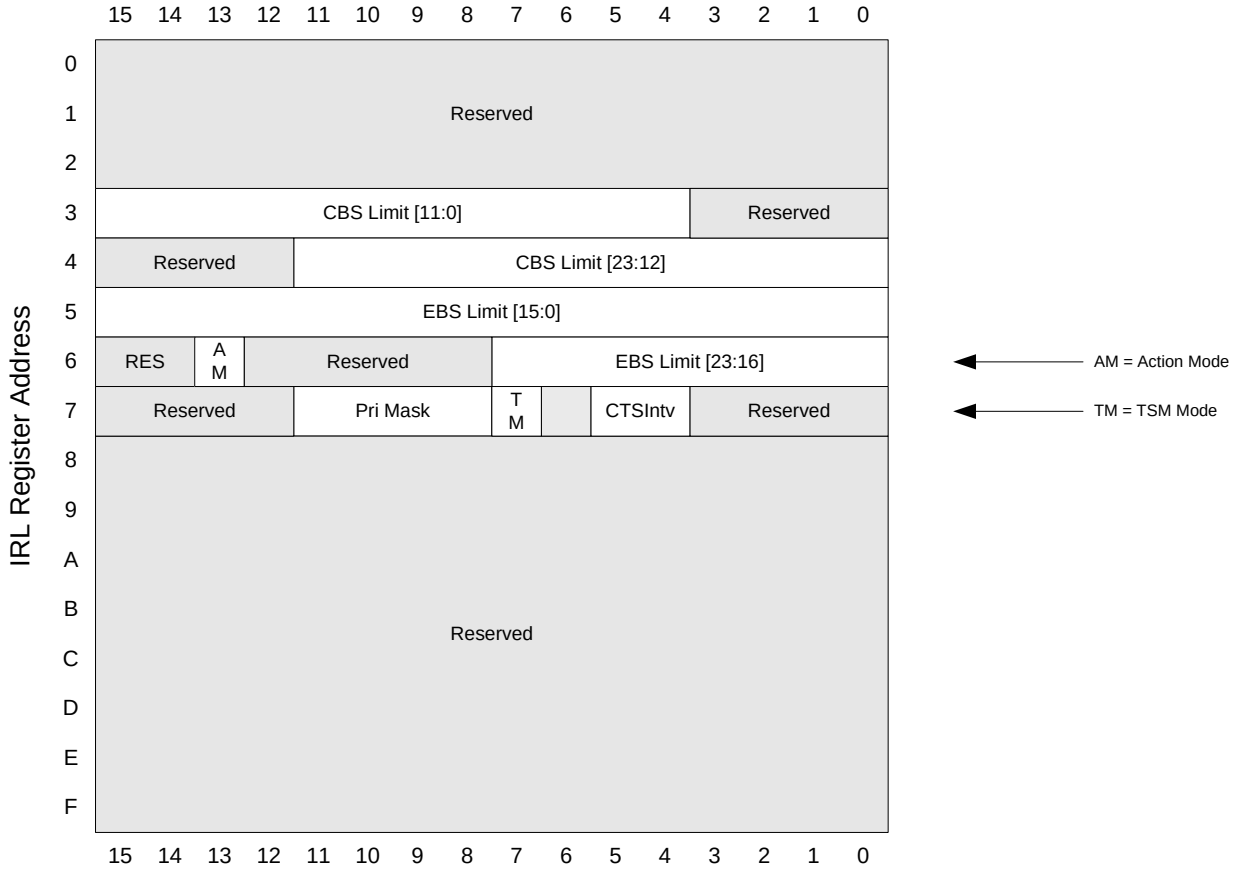


Table 223: IRL Bucket Configuration Register
Offset: 0x3 or decimal 3

Bits	Field	Type	Description
15:4	CBSLimit [11:0]	RWR	Committed Burst Size limit (lower 12 bits). This indicates the committed information burst amount. Set this value to the maximum number of AVB Layer 1 bytes per Observation Interval this AVB Class is not to exceed on average. AVB streams that exceed this rate on average will be discarded. The average rate is needed as bursting of AVB streams is allowed due to interfering frames. But the average should not be exceeded. See EBSLimit (IRL offset 0x07) to set a limit on the size of these AVB stream bursts. The rate limit is: Rate in bits/sec = (8*CBSLimit)/CTSIntv Where: CBSLimit is this register (IRL offset 0x03 & 0x04) CTSIntv is at IRL offset 0x07 The upper bits of this register is at IRL offset 0x04 (the next register below). In ITSM mode, CBS Limit must be smaller than EBS Limit (IRL offset 0x07).
3:0	Reserved	RES	Reserved for future use.

Table 224: IRL Bucket Configuration Register
Offset: 0x4 or decimal 4

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future uses.
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). When TSMMode is 0x1, the CBSLimit bits 23 down to 16 are reserved.

Table 225: IRL Bucket Configuration Register
Offset: 0x5 or decimal 5

Bits	Field	Type	Description
15:0	EBSLimit	RWR	Excess Burst Size limit (lower 16 bits). This indicates the excess information burst amount. Set this value to the maximum number of AVB Layer 1 bytes over multiple Observation Intervals this AVB Class is not to exceed ever. As AVB frames are allowed to burst to catch up due to interfering frames, this register allows for that catch up amount. AVB frames that exceed this rate will be discarded. Bursting support assumes Action Mode – IRL offset 0x06 – is cleared to a zero. The rate limit is: When ActionMode (below) = 1, EBSLimit is not used. When ActionMode (below) = 0, Token count can exceed CBSLimit (IRL offset 0x03 & 0x04) but cannot exceed this EBSLimit The upper bits of this register is at IRL offset 0x06 (the next register below)

Table 226: IRL Bucket Configuration Register
Offset: 0x6 or decimal 6

Bits	Field	Type	Description
15	Reserved	RWR	Reserved for future use.
13	Action Mode[1]	RWR	IRL 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.
10:8	Reserved	RES	Reserved for future uses.
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). See the description of these bits there. When TSMMode is 0x1, the EBSLimit bits 23 down to 16 are reserved.

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 227: IRL Bucket Configuration Register
Offset: 0x7 or decimal 7

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future uses.
11:8	PriMask	RWR	Priority Mask. For each PriMask bit: 0 = Don't account for this priority 1 = Account for this QPri AVB priority These bits must be used to select which AVB queue priority or priorities should be included in the rate limiting function of this resource. Bit 0 of this register points to priority 0, bit 1 points to priority 1, etc. For each these bits being set to a one, it indicates that this particular rate resource is slated to account for incoming frames with the enabled bits' QPri AVB priority.
7	TSM Mode	RWR	Time Slot Metering mode. This bit must be 0x1 for AVB/ITSM Rate Limiting. When 0x1, this field specifies to the hardware to operate the rate resource bucket in Time Slot Metering mode. In the mode the rate bucket doesn't get drained at the committed information rate but gets updated every Class Time Slot (CTS) Interval. The actual CTS interval values are defined in bits below in this register.
6	Reserved	RES	Reserved for future use.
5:4	CTS Intv	RWR	Class Time Slot Interval This is the measurement time interval over which various device logic operates including ingress and egress time slot metering logic. When 0x0, this field indicates that the Time Slot Metering interval is 62.5 us. When 0x1, this field indicates that the Time Slot Metering interval is 125 us. When 0x2, this field indicates that the Time Slot Metering interval is 250 us. When 0x3, this field indicates that the Time Slot Metering interval is 1 ms. **NOTE:** This timer operates off of a reference clock from the TAI block where each pulse is assumed to be 3.125 uSec to get the stated times (IRLGenAmt, IRLClkComp & IRLClkCompSubPs @ TAI offsets 0x06 to 0x08).
3:0	Reserved	RES	Reserved for future use.

1.5.1 AVB/TSN Registers

The device contains global registers that affect all the Audio Video Bridging (AVB) & Time Sensitive Networking (TSN) functions of the device. These registers are accessed using AVBCommand and AVBData registers which are found in Global 2 offset 0x16 & 0x17). The Global 2 AVB registers are used to access the following AVB blocks by using various AVBBlock values (Global 2 offset 0x16):

- 0x0 = 802.1AS Precise Timing Protocol (PTP) and Time Application Interface (TAI) registers
- 0x1 = 802.1BA Audio Video Bridging (AVB/TSN) Policy registers
- 0x2 = 802.1Qav per Class Shaping and Pacing registers
- 0x3 = 802.1Qbv per Queue Time Aware Shaper registers

1.5.2 PTP Registers

The following are the register bits used for configuration and status information to and from the software / CPU sub-system for Precise Time Protocol logic for audio-video bridging applications. These registers are accessed by using the AVB Command and AVB Data registers (Global 2 offset 0x16 and Offset 0x17).

Table 228: PTP Port Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x0 & AVBPort = 6:0)

PTP Port Register Data Bits

AVB Register Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	TransSpec				DT	Reserved									DO	DP	Configuration
1	RES	IP Jump						Reserved				ET Jump					Configuration
2	Arrival TS Mode							FA	Reserved						DE	AE	Configuration
3	Arrival LED Control							Departure LED Control									Configuration
4–7	Reserved																Configuration
8	Reserved														AT0	V0	Status
9	PTP Arrival 0 Time [15:0]																Status
A	PTP Arrival 0 Time [31:16]																Status
B	PTP Arrival 0 Sequence ID																Status
C	Reserved														AT1	V1	Status
D	PTP Arrival 1 Time [15:0]																Status
E	PTP Arrival 1 Time [31:16]																Status
F	PTP Arrival 1 Sequence ID																Status
10	Reserved														DT	TV	Status
11	PTP Departure Time [15:0]																Status
12	PTP Departure Time [31:16]																Status
13	PTP Departure Sequence ID																Status
14–1F	Reserved																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

DT = Disable Trans Spec Check
DO = Disable TS Overwrite
DP = Disable PTP

DE = PTP Departure Int Enable
AE = PTP Arrival Int Enable
FA = Filter 802.1 Act from LEDs

AT0 = PTP Arrival 0 Int Status
V0 = PTP Arrival 0 Time Valid

AT1 = PTP Arrival 1 Int Status
V1 = PTP Arrival 1 Time Valid

DT = PTP Departure Int Status
V1 = PTP Departure Time Valid

Table 229: PTP Port Config Register
Offset: 0x00 or decimal 0

Bits	Field	Type	Description
15:12	TransSpec	RWS 0x1	PTP Transport Specific value. The Transport Specific bits present in PTP Common header are used to differentiate between IEEE1588, IEEE802.1AS etc. frames. This is to differentiate between various timing protocols running on either Layer2 or higher protocol layers. In hardware, in addition to comparing the EtherType to determine that the incoming frame is a PTP frame, it compares the TransSpec bits to the incoming PTP common headers' Transport Specific bits. If there is a match then hardware logic time stamps the frames indicated by MsdIDTSEn and optionally interrupts the CPU. If there were to be no match then the hardware wouldn't perform any operations in the PTP core. For IEEE 1588 networks this is expected to be configured to a 0x0 and for IEEE 802.1AS networks this is expected to be configured to 0x1.
11	DisTSpec Check	RWR	Disable Transport Specific Check. 0x1 = Disables checking for Transport Specific part of the PTP Common header between the incoming packet data and the configured TransSpec (PTP Port Config, Offset ox0). 0x0 = Enabled checking for Transport Specific part of the PTP Common header between the incoming packet data and the configured TransSpec (PTP Port Config, Offset ox0).
10:2	Reserved	RES	Reserved for future use.
1	DisTS Overwrite	RWR	Disable Time Stamp Counter Overwriting. When set to 0x1, PTPArr0Time, PTPArr1Time and PTPDepTime values don't get overwritten till their corresponding valid bits (defined in PTP Port Status Data Structure below), are not cleared. This situation only arises when a port based time stamp counter is written by hardware logic but software layer hasn't read the data. When set to 0x0, PTPArr0Time, PTPArr1Time and PTPDepTime values do get overwritten even though their corresponding valid bits (defined in PTP Port Status Data Structure below), are not cleared.
0	DisPTP	RWR	Disable Precise Time Stamp logic (per-port bit). 0x0 = PTP logic within the chip is enabled. 0x1 = PTP logic is disabled i.e., hardware logic doesn't recognize or timestamp PTP frames. Even interrupt generation logic is disabled.

Table 230: PTP Port Config Register
Offset: 0x01 or decimal 1

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:8	IPJump	RWS 0x2	Internet Protocol Jump added to ETJump below. Set this register to point to the start of the frame's IP Version byte (802.1Q tagged frames are automatically compensated by ETJump). This field specifies to the PTP hardware logic how many bytes should it skip starting at the value specified by ETJump (PTP Port Config register, Offset 0x1). Note that this specifies the jump to the beginning of the IPv4 or IPv6 headers for the hardware parser. This allows flexibility in the hardware to skip past the protocol chains that are specific to customer networks including MPLS etc. For example if ETJump is programmed to 0xC and IPJump is programmed to 0x16, this indicates to hardware to skip 0x22 bytes in order to get to the IP header. It can either be IPv4 or IPv6 header. **NOTE:** A value of 0x0 or 0x1 in this register is invalid and is not supported.
7:5	Reserved	RES	Reserved for future use.
4:0	ETJump	RWS 0xC	EtherType Jump points to the start of the frame's EtherType (assuming it is not 802.1Q tagged). This field specifies to the PTP hardware logic how many bytes should it skip starting from MAC-DA of the frame to get to the EtherType of the frame. The hardware would skip so many bytes and then compare the next 2 bytes to the configured PTPEType (PTP Global Config Register, Offset 0x0). If the PTPEType value (PTP Global offset 0x00) is found as the Ether Type in the frame, Layer 2 PTP processing occurs. If 0x0800 or 0x86DD is found the Layer 4 PTP processing occurs. If none of these values is found, PTP frame decoding is passed to the IPJump field above. This allows flexibility in the hardware to skip past the protocol chains that are specific to customer networks including DSA-Tag, IEEE802.1Q tag, Provider tag etc.

Table 231: PTP Port Config Register
Offset: 0x02 or decimal 2

Bits	Field	Type	Description
15:8	ArrTSMode	RWR	Arrival Time Stamp Mode. This field is used to configure the Arrival Time Stamp frame modification mode as follows: 0x00 = Arrival Time Stamp frame modification is disabled. 0x01 to 0x0F = Reserved for future use. 0x10 = Overwrite the PTPArr0Time associated with enabled PTP Event frames into the frame in the four Reserved bytes in the PTP Common Header. 0x11 to 0xFF = Reserved for future use. **NOTE:** Changes to this register will take effect only when the receive port is idle or the port is in the Disabled Port State (Port offset 0x04). **NOTE:** Added PTPArr0Time bytes that increase the frame's size are included in the MIB counters and policy is performed on the resulting frame (e.g., TCAM matching and frame size checking).
7	FilterAct	RWR	Filter LED Activity. Filter all or most of the 802.1 Protocol frames from the Port's Activity LEDs. When this bit is set to a one all 802.1 protocol frames (those with a DA = 01:C2:80:00:00:0x) will be potentially filtered from the port's Activity LED as determined by the ArrLEDCtrl and DepLEDCtrl registers (PTP Port offset 0x03). When this bit is cleared to a zero only the 802.1 gPTP protocol frames will be potentially filtered from the port's Activity LED. **NOTE:** This bit affects the port's LEDs only. The frames themselves are not filtered inside the switch because of this bit.
6:2	Reserved	RES	Reserved for future use.
1	PTPDepInt En	RWR	Precise Time Protocol Port Departure Interrupt enable. 0 = Disable PTP Departure capture interrupts 1 = Enable PTP Departure capture interrupts This field indicates the per-port interrupt enable for outgoing PTP frame from a given port. When a bit is enabled in this field it indicates that whenever hardware logic time stamps a PTP frame to this port, it needs to send an interrupt to the CPU. **NOTE:** Hardware logic only time stamps the PTP frames when configured to do so by MsdIDTSEn field (specified above).

Table 231: PTP Port Config Register (Continued)
Offset: 0x02 or decimal 2

Bits	Field	Type	Description
0	PTPArrInt En	RWR	Precise Time Protocol Port Arrival Interrupt enable. 0 = Disable PTP Arrival capture interrupts 1 = Enable PTP Arrival capture interrupts This field indicates the per-port interrupt enable for incoming PTP frames from a given port. When a bit is enabled in this field it indicates that whenever hardware logic time stamps a PTP frame from this port, it needs to send an interrupt to the CPU. **NOTE:** Hardware logic only time stamps the PTP frames when configured to do so by MsdIDTSEn field (specified above).

Table 232: PTP Port Config Register
Offset: 0x03 or decimal 3

Bits	Field	Type	Description
15:8	ArrLED Ctrl	RWR	LED control for packets entering the device. When 0x0, if a received frame is classified as a PTP frame or an 802.1 protocol frame, the LED does not blink. But it blinks for every non-PTP or non-802.1 protocol frame. When 0x1, the LED blinks for every received frame classified as a PTP frame or an 802.1 protocol frame. It also blinks for every non-PTP frame or non-802.1 protocol frame. When 0xn, the LED blinks once for every n received frames classified as PTP or 802.1 protocol. It also blinks for every non-PTP frame or non-802.1 protocol frame. **NOTE:** This tracks all received PTP frames (even though the PTP core time stamps only the PTP event messages) and not just PTP frames that need time stamping or it tracks all received 802.1 protocol frames (any frame with a DA = 01:C2:80:00:00:0x). The FilterAct bit in PTP Port offset 0x02 controls which frames are tracked. This register affect the port's Activity LED only. It does not change how the frames progress through the switch.
7:0	DepLED Ctrl	RWS 0x80	LED control for packets departing the device. When 0x0, if a transmitting frame is classified as a PTP frame or an 802.1 protocol frame, the LED does not blink. But it blinks for every non-PTP frame or non-802.1 protocol. When 0x1, the LED blinks for every transmitting frame classified as a PTP frame or an 802.1 protocol frame. It also blinks for every non-PTP frame or non-802.1 protocol frame. When 0xn, the LED blinks once for every n transmitting frames classified as PTP or 802.1 protocol. It also blinks for every non-PTP frame or non-802.1 protocol frame. **NOTE:** This tracks all transmitting PTP frames (even though the PTP core time stamps only the PTP event messages) and not just PTP frames that need time stamping or it tracks all transmitted 802.1 protocol frames (any frame with a DA = 01:C2:80:00:00:0x). The FilterAct bit in PTP Port offset 0x02 controls which frames are tracked. This register affect the port's Activity LED only. It does not change how the frames progress through the switch.

Table 233: PTP Port Status Register
Offset: 0x08 or decimal 8

Bits	Field	Type	Description
15:3	Reserved	RES	Reserved for future use.
2:1	PTPArr0IntStatus	RWR	Precise Time Protocol Arrival Time 0 Interrupt Status The PTP Arrival time 0 Interrupt bit gets set for a given port when an incoming PTP frame is time stamped in PTPArr0Time counter. 0x0 = Normal i.e., none of the error conditions stated below are valid for this packet. 0x1 = If the PTPArr0Time counter with its associated valid and SequenceID got overwritten as there were more than one PTP frame that needed to use arrival0 counters arrived into the switch through this port before CPU cleared the corresponding valid and counter bits for the previous PTP frame(s). 0x2 = If the incoming frame couldn't be time stamped in hardware because the DisTSOverwrite was set to a 0x1 and PTPArr0TimeValid was 0x1 when this PTPFrame was processed in hardware logic. This can happen when there is more than one PTP frame that needs time stamping into arrival 0 counters arrives into the switch core before CPU clears the valid bits for the previous frame. 0x3 = Reserved **NOTE:** If the PTP frame gets discarded inside the switch for policy, CRC, queue congestion or any other reasons then one of the PTP arrival discard counters get updated (PTPNonTSArrDisCtr or PTPTSArrDisCtr). See the discard counter description for further details.
0	PTPArr0 TimeValid	RWR	Precise Time Protocol Arrival 0 Time Valid When the PTPArr0Time value is updated by hardware, this bit is set to a 0x1 validating the time counter. 0x0 = PTPArr0Time is not valid. 0x1 = PTPArr0Time is valid and PTPArr0IntStatus represents the status information for the PTPArr0Time counter. Note that this is set by hardware for the frames which are assured to reach the CPU. For frames with CRC error etc., this bit will not be set but either PTPNonTSArrCtr or PTPTSArrCtr is updated. Note that this valid bit needs to be cleared by software after reading the value and hardware doesn't provide any auto-clearing mechanisms. This is because hardware has no way to figure out if software is done reading all the relevant registers for a particular Time counter before clearing the valid bit.

Note: PTP Port register offsets 0x08 to 0x0B are a Capture Group. If an AVBOp of 0x6 is used (Read with Pos– Increment - Global 2 offset 0x16) with an AVBAddr within this Capture Group's address range, the current value of the AVBAddr register along with the values in the next 3 registers (AVBAddr+1, AVBAddr+2 and AVBAddr+3) are saved in holding registers. Up to the next four reads of the AVBData register (Global 2 offset 0x17) will return this captured data insuring the integrity of these registers between the separate reads. Reads beyond four will return non-captured, real time data. Any other AVBOp will terminate the current captured read function as well. A new capture will only occur if another AVBOp of 0x6 is used with an AVBAddr within a Capture Group's address range.

Table 234: PTP Port Status Register
Offset: 0x09 or decimal 9

Bits	Field	Type	Description
15:0	PTPArr0 Time [15:0]	RWR	Precise Time Protocol Arrival 0 Time counter bits 15 to 0 of a 32-bit register. This indicates the PTP Arrival 0 time stamp value that is captured by the PTP logic for a PTP frame that needs to be time stamped. The captured time stamp value is from a Global Timer counter running off of a switch internal clock. The value in this counter is validated by PTPArr0TimeValid bit and PTPArr0IntStatus indicates the status of the PTP frame through the device as described above. **NOTE:** The maximum jitter associated with time stamping within the hardware is one TSClkPer amount. The upper 16-bits of this register are contained in the register below.

Table 235: PTP Port Status Register
Offset: 0x0A or decimal 10

Bits	Field	Type	Description
15:0	PTPArr0 Time [31:16]	RWR	Precise Time Protocol Arrival 0 Time counter bits 31 to 16 of a 32-bit register. This indicates the PTP Arrival 0 time stamp value that is captured by the PTP logic for a PTP frame that needs to be time stamped. The captured time stamp value is from a Global Timer counter running off of a switch internal clock. The value in this counter is validated by PTPArr0TimeValid bit and PTPArr0IntStatus indicates the status of the PTP frame through the device as described above. **NOTE:** The maximum jitter associated with time stamping within the hardware is one TSClkPer amount. The lower 16-bits of this register are contained in the register above.

Table 236: PTP Port Status Register
Offset: 0x0B or decimal 0x11

Bits	Field	Type	Description
15:0	PTPArr0 SeqId	RWR	Precise Time Protocol Arrival 0 Sequence Identifier. This indicates the sequence identifier (extracted in hardware from incoming frames PTPCommonHeader) for the frame whose time stamp information has been captured by hardware logic in PTPArr0Time register.

Table 237: PTP Port Status Register
Offset: 0x0C or decimal 12

Bits	Field	Type	Description
15:3	Reserved	RES	Reserved for future use.
2:1	PTPArr1IntStatus	RWR	Precise Time Protocol Arrival Time 1 Interrupt Status The PTP Arrival time 1 Interrupt bit gets set for a given port when an incoming PTP frame is time stamped in PTPArr1Time counter. 0x0 = Normal i.e., none of the error conditions stated below are valid for this packet. 0x1 = If the PTPArr1Time counter with its associated valid and SequenceID got overwritten as there were more than one PTP frame that needed to use arrival1 counters arrived into the switch through this port before CPU cleared the corresponding valid and counter bits for the previous PTP frame(s). 0x2 = If the incoming frame couldn't be time stamped in hardware because the DisTSOverwrite was set to a 0x1 and PTPArr1TimeValid was 0x1 when this PTPFrame was processed in hardware logic. This can happen when there is more than one PTP frame that needs time stamping into arrival 1 counters arrives into the switch core before CPU clears the valid bits for the previous frame. 0x3 = Reserved **NOTE:** If the PTP frame gets discarded inside the switch for policy, CRC, queue congestion or any other reasons then one of the PTP arrival discard counters get updated (PTPNonTSArrDisCtr or PTPTSArrDisCtr). See the discard counter description for further details.

Table 237: PTP Port Status Register (Continued)
Offset: 0x0C or decimal 12

Bits	Field	Type	Description
0	PTPArr1 TimeValid	RWR	Precise Time Protocol Arrival 1 Time Valid When the PTPArr1Time value is updated by hardware, this bit is set to a 0x1 validating the time counter. 0x0 = PTPArr1Time is not valid. 0x1 = PTPArr1Time is valid and PTPArr1IntStatus represents the status information for the PTPArr1Time counter. Note that this is set by hardware for the frames which are assured to reach the CPU. For frames with CRC error etc., this bit will not be set but either PTPNonTSArrCtr or PTPTSArrCtr is updated. **NOTE:** This valid bit needs to be cleared by software after reading the value and hardware doesn't provide any auto-clearing mechanisms. This is because hardware has no way to figure out if software is done reading all the relevant registers for a particular Time counter before clearing the valid bit.

Note: PTP Port register offsets 0x08 to 0x0B are a Capture Group. If an AVBOp of 0x6 is used (Read with Pos– Increment - Global 2 offset 0x16) with an AVBAddr within this Capture Group's address range, the current value of the AVBAddr register along with the values in the next 3 registers (AVBAddr+1, AVBAddr+2 and AVBAddr+3) are saved in holding registers. Up to the next four reads of the AVBData register (Global 2 offset 0x17) will return this captured data insuring the integrity of these registers between the separate reads. Reads beyond four will return non-captured, real time data. Any other AVBOp will terminate the current captured read function as well. A new capture will only occur if another AVBOp of 0x6 is used with an AVBAddr within a Capture Group's address range.

Table 238: PTP Port Status Register
Offset: 0x0D or decimal 13

Bits	Field	Type	Description
15:0	PTPArr1 Time [15:0]	RWR	Precise Time Protocol Arrival 1 Time counter bits 15 to 0 of a 32-bit register. This indicates the PTP Arrival 1 time stamp value that is captured by the PTP logic for a PTP frame that needs to be time stamped. The captured time stamp value is from a Global Timer counter running off of a switch internal clock. The value in this counter is validated by PTPArr1TimeValid bit and PTPArr1IntStatus indicates the status of the PTP frame through the device as described above. **NOTE:** The maximum jitter associated with time stamping within the hardware is one TSClkPer amount. The upper 16-bits of this register are contained in the register below.

Table 239: PTP Port Status Register
Offset: 0x0E or decimal 14

Bits	Field	Type	Description
15:0	PTPArr1 Time [31:16]	RWR	Precise Time Protocol Arrival 1 Time counter bits 31 to 16 of a 32-bit register. This indicates the PTP Arrival 1 time stamp value that is captured by the PTP logic for a PTP frame that needs to be time stamped. The captured time stamp value is from a Global Timer counter running off of a switch internal clock. The value in this counter is validated by PTPArr1TimeValid bit and PTPArr1IntStatus indicates the status of the PTP frame through the device as described above. **NOTE:** The maximum jitter associated with time stamping within the hardware is one TSClkPer amount. The lower 16-bits of this register are contained in the register above.

Table 240: PTP Port Status Register
Offset: 0x0F or decimal 0x15

Bits	Field	Type	Description
15:0	PTPArr1 SeqId	RWR	Precise Time Protocol Arrival 1 Sequence Identifier. This indicates the sequence identifier (extracted in hardware from incoming frames PTPCommonHeader) for the frame whose time stamp information has been captured by hardware logic in PTPArr1Time register.

Table 241: PTP Port Status Register
Offset: 0x10 or decimal 16

Bits	Field	Type	Description
15:3	Reserved	RES	Reserved for future use.
2:1	PTPDepIntStatus	RWR	Precise Time Protocol Departure Time Interrupt Status The PTP Departure time Interrupt bit gets set for a given port when an incoming PTP frame is time stamped in PTPDepTime counter. 0x0 = Normal i.e., none of the error conditions stated below are valid for this packet. 0x1 = If the PTPDepTime counter with its associated valid and SequenceID got overwritten as there were more than one PTP frame that needed to use departure counter departed out of the switch through this port before CPU cleared the corresponding valid and counter bits for the previous PTP frame(s). 0x2 = If the outgoing frame couldn't be time stamped in hardware because the DisTSOverwrite was set to a 0x1 and PTPDepTimeValid was 0x1 when this PTPFrame was processed in hardware logic. This can happen when there is more than one PTP frame that needs time stamping into departure counter leaves the switch core before CPU clears the valid bits for the previous frame. 0x3 = Reserved **NOTE:** If the PTP frame gets discarded inside the switch for CRC reasons then the PTP departure discard counter gets updated (PTPNonTSDepDisCtr or PTPTSDepDisCtr). See the discard counter description for further details.

Table 241: PTP Port Status Register (Continued)
Offset: 0x10 or decimal 16

Bits	Field	Type	Description
0	PTPDep TimeValid	RWR	Precise Time Protocol Departure Time Valid When the PTPDepTime value is updated by hardware, this bit is set to a 0x1 validating the time counter. 0x0 = PTPDepTime is not valid. 0x1 = PTPDepTime is valid and PTPDepIntStatus represents the status information for the PTPDepTime counter. Note that this is set by hardware for the frames which are assured to depart the port. For frames with CRC error etc., this bit will not be set but either PTPNonTSDepCtr or PTPTSDepCtr is updated. **NOTE:** This valid bit needs to be cleared by software after reading the value and hardware doesn't provide any auto-clearing mechanisms. This is because hardware has no way to figure out if software is done reading all the relevant registers for a particular Time counter before clearing the valid bit.

Note: PTP Port register offsets 0x08 to 0x0B are a Capture Group. If an AVBOp of 0x6 is used (Read with Pos– Increment - Global 2 offset 0x16) with an AVBAddr within this Capture Group's address range, the current value of the AVBAddr register along with the values in the next 3 registers (AVBAddr+1, AVBAddr+2 and AVBAddr+3) are saved in holding registers. Up to the next four reads of the AVBData register (Global 2 offset 0x17) will return this captured data insuring the integrity of these registers between the separate reads. Reads beyond four will return non-captured, real time data. Any other AVBOp will terminate the current captured read function as well. A new capture will only occur if another AVBOp of 0x6 is used with an AVBAddr within a Capture Group's address range.

Table 242: PTP Port Status Register
Offset: 0x11 or decimal 17

Bits	Field	Type	Description
15:0	PTPDep Time [15:0]	RWR	Precise Time Protocol Departure Time counter bits 15 to 0 of a 32-bit register. This indicates the PTP Departure time stamp value that is captured by the PTP logic for a PTP frame that needs to be time stamped. The captured time stamp value is from a Global Timer counter running off of a switch internal clock. The value in this counter is validated by PTPDepTimeValid bit and PTPDepIntStatus indicates the status of the PTP frame through the device described above. **NOTE:** The maximum jitter associated with time stamping within the hardware is one TSClkPer amount. The upper 16-bits of this register are contained in the register below.

Table 243: PTP Port Status Register
Offset: 0x12 or decimal 18

Bits	Field	Type	Description
15:0	PTPDep Time [31:16]	RWR	Precise Time Protocol Departure Time counter bits 31 to 16 of a 32-bit register. This indicates the PTP Departure time stamp value that is captured by the PTP logic for a PTP frame that needs to be time stamped. The captured time stamp value is from a Global Timer counter running off of a switch internal clock. The value in this counter is validated by PTPDepTimeValid bit and PTPDepIntStatus indicates the status of the PTP frame through the device described above. **NOTE:** The maximum jitter associated with time stamping within the hardware is one TSClkPer amount. The lower 16-bits of this register are contained in the register above.

Table 244: PTP Port Status Register
Offset: 0x13 or decimal 0x19

Bits	Field	Type	Description
15:0	PTPDep SeqId	RWR	Precise Time Protocol Departure Sequence Identifier. This indicates the sequence identifier (extracted in hardware from incoming frames PTPCommonHeader) for the frame whose time stamp information has been captured by hardware logic in PTPDepTime register.

Table 245: PTP Global Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x0 & AVBPort = 0x1F)

PTP Global Register Data Bits

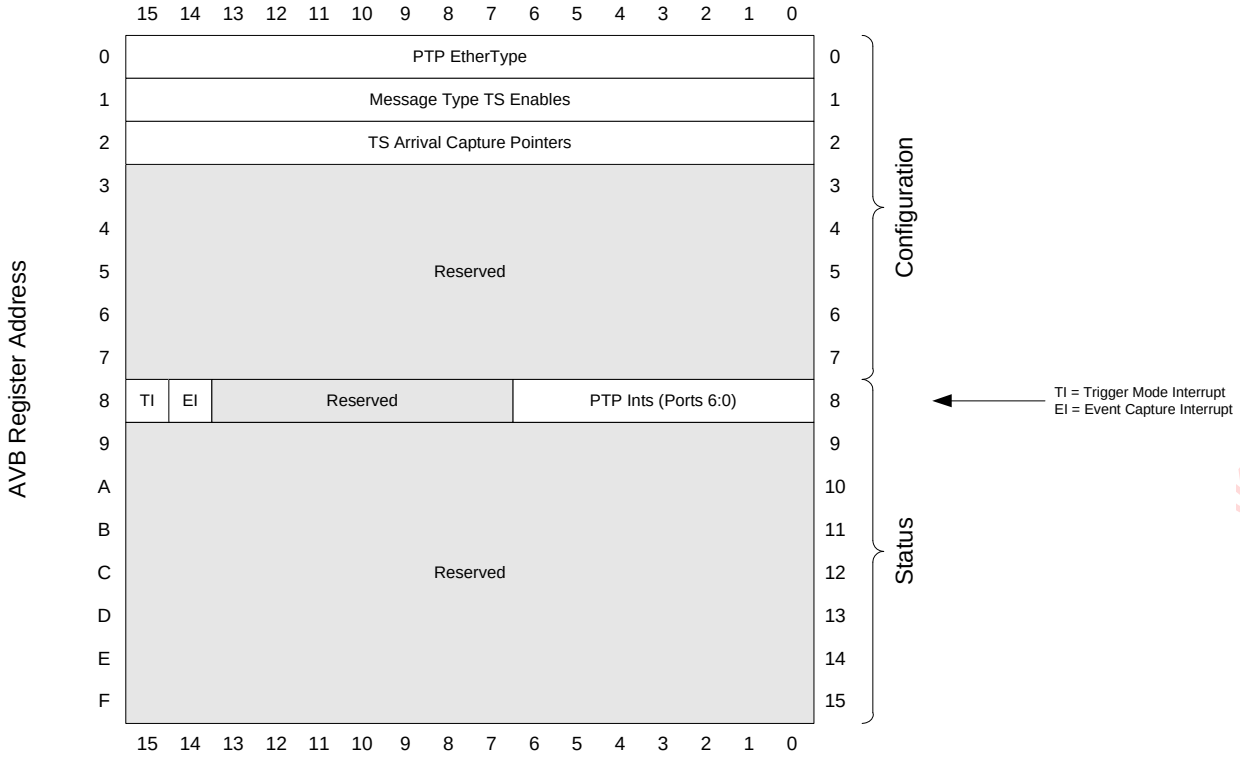


Table 246: PTP Global Config, AVBPort = 0x1F
Offset: 0x00 or decimal 0 – PTP Global

Bits	Field	Type	Description
15:0	PTPEType	RWS to 0x88F7	Precise Time Protocol EtherType. All layer 2 PTP frames are recognized using a combination of a specific EtherType and MessageType values (part of the PTP Common Header). This field is used to identify the EtherType on these frames. The MsgTypeEn (specified below in offset 0x01) qualifies the types of frames that the hardware needs to time stamp. For IEEE 802.1AS and IEEE1588 over Layer 2 Ethernet, the EtherType is expected to be programmed to 0x88F7.

Table 247: PTP Global Config, AVBPort = 0x1F
Offset: 0x01 or decimal 1 – PTP Global

Bits	Field	Type	Description
15:0	MsgType En	RWR	Message Type Time Stamp Enable. MessageType is part of the PTP common header. There are PTP frames which need to be time stamped and some that don't need to be. This field identifies the PTP frame types that need to be time stamped. The MessageType read from PTP frames is vectorized[1] and then used to access the appropriate bit in this register. If the selected bit is set to a one then that frame type will be time stamped both in ingress and egress. Else that frame type will not be time stamped. For example if MessageType field (in the PTP common header) with a value of 0x4 needs to be time stamped in hardware then MsgTypeEn[4] should be set to a one. Then the incoming PTP frames with a MessageTypefield of 0x4 will get time stamped into one of the Port's two available arrival capture registers (either PTPArr0Time or PTPArr1Time as identified by TSArrPtr[4] bit below). All outgoing PTP frames with the MessageType field of 0x4 will be timestamped into the Port's PTPDepTime capture register.

1. Vectorized term here refers to converting the hexadecimal MessageType field into a sixteen bit binary number.

Table 248: PTP Global Config, AVBPort = 0x1F
Offset: 0x02 or decimal 2 – PTP Global

Bits	Field	Type	Description
15:0	TSArrPtr	RWR	Time Stamp Arrival Time Capture Pointer. If the incoming PTP frame needs to be time stamped (based on MsgTypeEn), this field determines whether the hardware logic should use PTPArr0Time or PTPArr1Time for storing the arriving frames' time stamp information. This field corresponds to the sixteen combinations of the vectorized MessageType. For example if TSArrPtr[2] is set to a one it indicates to the hardware that if MsgTypeEn [2] is set to a one then PTP frames with MessageType = 0x2 will use PTPArr1Time counter for storing the incoming PTP frames' time stamp. On the contrary if TSArrPtr[2] is cleared to a zero that indicates to the hardware that if MsgTypeEn[2] is set to a one then PTP frames with MessageType = 0x2 will use PTPArr0Time counter for storing the incoming PTP frames' time stamp.

Table 249: PTP Status, AVBPort = 0x1F
Offset: 0x08 or decimal 8 – PTP Global

Bits	Field	Type	Description
15	TrigGen Int	ROC	Trigger generate mode Interrupt. The TrigGenInt bit gets set by the TAI block when the TrigGenIntEn is set to a one (TAI offset 0x00) and when the one shot pulse is generated (TrigMode is set to one in TAI offset 0x00). It gets cleared by the reading of this register. This interrupt gets tied to the AVBInt interrupt (Global 1 offset 0x00).
14	Event Int	RO	Event Capture Interrupt. This bit gets set by the TAI block when the EventIntEn is set to a one (TAI offset 0x00) and when an EventReq is captured in the EventCap Register. It gets cleared by writing a zero to the EventCapValid register (TAI offset 0x09). **NOTE:** This interrupt bit will also be set to a one, if enabled, whenever the EventCapValid bit (TAI offset 0x09) is set to a one. This allows software to test its interrupt routine. This interrupt gets tied to the AVBInt interrupt (Global 1 offset 0x00).
13:7	Reserved	RES	Reserved for future use.
6:0	PTPInt	RO	Precise Time Protocol Interrupts. These PTP Interrupt bits gets set for a given port when an incoming PTP frame is time stamped and PTPArrIntEn for that port is set to a one. Similarly the PTP Interrupt bits get set for a given port when an outgoing PTP frame is time stamped and PTPDepIntEn for that port is set to a one. The hardware logic sets this per port bit based on above criteria and gets cleared upon software reading and clearing the corresponding time counter valid bits that are valid for that port. Bit 0 is for Port 0, bit 1 is for Port 1, etc. These interrupts are connected to the AVBInt interrupt (Global 1 offset 0x00).

Table 250: PTP TAI Global Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x0 & AVBPort = 0x1E)

PTP TAI Global Register Data Bits

AVB Register Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	CO	CS	EP	TP		IR	Int Enables		TL	Trig Lock Range			BU	MS	TM	TR	Configuration
1	TS Clock Period																Configuration
2	Trig Generation Amount [15:0]																Configuration
3	Trig Generation Amount [31:16]																Configuration
4	TD	Trig Clock Compensation															Configuration
5	Pulse Width						Pulse Width Range		Trig Clock Comp Sub ps								Configuration
6	IRL Clk Generation Amount [15:0]																Configuration
7	ID	IRL Clock Compensation															Configuration
8	Reserved								IRL Clock Comp Sub ps								Configuration
9		CT	Reserved				CE	CV	Event Capture Counter								Status
A	Event Capture Time [15:0]																Status
B	Event Capture Time [31:16]																Status
C	Reserved																Status
D	Reserved																Status
E	PTP Global Time [15:0]																Status
F	PTP Global Time [31:16]																Status
10	Trig Generation Time [15:0]																Configuration
11	Trig Generation Time [31:16]																Configuration
12	Reserved												LV	Lock Correction			Configuration
13	Reserved																Configuration
14	Reserved																Configuration
15	Reserved																Configuration
16	Reserved																Configuration
17	Reserved																Configuration
18	Reserved																Configuration
19	Reserved																Configuration
1A	Reserved																Configuration
1B	Reserved																Configuration
1C	Reserved																Configuration
1D	Reserved																Configuration
1E		PE				Sec Rec Clk Sel			RES					Pri Rec Clk Sel			Configuration
1F	Reserved																Configuration
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

CO = Event Capture Overwrite
CS = Event Counter Start
EP = Event Phase
TP = Trigger Phase
IR = IRL Clk Generation Request
TL = Trigger Lock Enable
BU = Block Update
MS = Multi PTP Sync Mode
TM = Trigger Mode
TR = Trigger Generation Request

TD = Trig Clock Comp Direction

ID = IRL Clock Comp Direction

CT = Capture Trigger Events
CE = Event Capture Error
CV = Event Capture Valid

LV = Lock Correction Valid

PE = PTP Ext Clk Sel

Table 251: TAI Global Config, AVBPort = 0x1E
Offset: 0x00 or decimal 0 – TAI

Bits	Field	Type	Description
15	Event CapOv	RWR	Event Capture Overwrite. 0 = Capture & Hold first PTP Event 1 = Capture all PTP Events & Retain the last PTP Event When this bit is cleared to a zero it configures the hardware to capture a single event, i.e., take a snapshot of PTP Global Timer value at the first EventReq (see EventPhase below) and wait for software to read the EventCapRegister before capturing another event. This mode returns the data from the first EventReq. When this bit is set to a one it enables overwriting the EventCap registers (TAI offsets 0x09 to 0x0B) whenever an EventReq occurs (see EventPhase below). In this mode the hardware will overwrite the EventCapRegister even if the previously captured event register data has not been read by the software. This mode returns the data from the last EventReq.
14	EventCtr Start	RWR	Event Counter Start. 0 = Do not increment the Event Capture Counter 1 = Increment the Event Capture Counter on EventReq's When this bit is cleared to zero the EventCapCtr is not modified even when EventReq occurs (see EventPhase below). When this bit is set to a one it enables incrementing the EventCapCtr register (TAI offset 0x09) whenever an EventReq occurs (see EventPhase below).
13	Event Phase	RWR	Event Phase. 0 = Event Requests occur on the rising edge of the PTP_EVREQ GPIO pin 1 = Event Requests occur on the falling edge of the PTP_EVREQ GPIO pin When this bit is cleared to a zero an EventReq occurs on the rising edge of the PTP_EVREQ input selected by the GPIO matrix (Global 2 offset 0x1A) or on the leading edge of PTP_TRIG when internally sampled (see the CaptureTrig bit in TAI offset 0x09). When this bit is set to a one an EventReq occurs on the falling edge of the PTP_EVREQ input selected by the GPIO matrix (Global 2 offset 0x1A) or on the trailing edge of PTP_TRIG when internally sampled. When PTP_TRIG is selected to be internally captured (instead of using the PTP_EVREQ GPIO pin – see Capture Trig in TAI offset 0x09) this Event Phase is used to invert the value of the normal internal PTP_TRIG that is captured. When Event Phase = 0 the leading edge (or normal rising edge) of the internal PTP_TRIG is captured. When Event Phase = 1 the trailing edge (or normal falling edge) of the internal PTP_TRIG is captured.

Table 251: TAI Global Config, AVBPort = 0x1E
Offset: 0x00 or decimal 0 – TAI

Bits	Field	Type	Description
12	TrigPhase	RWR	Trigger Phase. 0 = The PTP Trigger output is active high on the PTP_TRIG GPIO pin 1 = The PTP Trigger output is active low on the PTP_TRIG GPIO pin When this bit is cleared to a zero the active phase of the PTP_TRIG output***s*** to the GPIO matrix (Global 2 offset 0x1A) is***/are*** normal active high. For example, the pulse mode of PTP_TRIG will be normally low with a high pulse and the 50% duty cycle's leading edge is the rising edge. When this bit is set to a one the active phase of the PTP_TRIG output***s*** to the GPIO matrix (Global 2 offset 0x1A) is***/are*** inverted to be active low. For example, the pulse mode of PTP_TRIG will be normally high with a low pulse and the 50% duty cycle's leading edge is the falling edge. **NOTE:** This bit has no effect on the internal phase of PTP_TRIG or any other signal used in the internal blocks of the device.
11	Reserved	RES	Reserved for future use.
10	IRLClk Gen Req	RWS	Ingress Rate Limiter's Clock Generation Request/Enable. When this bit is set to a one, it enables a 50% duty cycle clock generation for the Ingress Rate Limiter's logic (IRL Clk) as configured by the IRLClkGenAmt, IRLClkComp (both ps & Sub ps) and IRLGenTime fields. On Reset, this IRL Clock defaults to a 3.125 uSec rate. This rate can be changed at any time by updating the IRLClkGenAmt &/or IRLClkComp (ps & Sub ps) registers.
9	TrigGen IntEn	RWR	Trigger Generator Interrupt Enable. 0 = Mask interrupts generated by the PTP Trigger logic 1 = Enable interrupts generated by the PTP Trigger logic When this bit is cleared to zero no interrupts are generated by the TrigGen logic. When this bit is set to a one the TAI block will generate an interrupt whenever a TrigGen pulse event has occurred. This interrupt will appear in the Trigger Mode Interrupt in PTP Global offset 0x08 where it will appear in the AVB Interrupt (Global 1 offset 0x00).

Table 251: TAI Global Config, AVBPort = 0x1E
Offset: 0x00 or decimal 0 – TAI

Bits	Field	Type	Description
8	EventCap IntEn	RWR	Event Capture Interrupt Enable. 0 = Mask interrupts generated by the PTP Event Capture logic 1 = Enable interrupts generated by the PTP Event Capture logic When this bit is cleared to zero no interrupts are generated by the EventCap logic. When this bit is set to a one the TAI block will generate an interrupt whenever an EventReq occurs. This interrupt will appear in the Event Capture Interrupt in PTP Global offset 0x08 where it will appear in the AVB Interrupt (Global 1 offset 0x00).
7	TrigLock	SC	Trigger Lock. When this bit is set to a one the leading edge of PTP_TRIG (see TrigPhase above) will be adjusted to the value contained in the TrigGenTime register (TAI offsets 0x10 & 0x11) if and only if the leading edge of PTP_TRIG occurs +/- the number of PTP Clocks as defined in the TrigLockRange register below and PTP_TRIG is enabled (TrigGenReq, below, is set to a one) and the TrigGenTime register is non-zero. Note: The TrigLockRange, the TrigGenTime registers must be configured before this bit is set to a one. Once the TrigGenTime past in time, this bit will self clear (i.e., it will be active for only one possible correction per wrap around of the 32-bit Global Timer). This bit will clear if the Global time has passed even if a correction was not needed or done. When this bit clears the Lock correction amount, if any, will be registered in the Lock Correction fields for PTP_TRIG (TAI offset 0x12).
6:4	TrigLock Range	RWR	Trigger Locking Range. These bits are used along with the TrigLock bit above. They determine the +/- error limit to adjust and re-center the leading edge of PTP_TRIG in PTP_CLK increments (8ns if using the internal clock) or PTP_EXTCLK increments (if using an external PTP clock).

Table 251: TAI Global Config, AVBPort = 0x1E
Offset: 0x00 or decimal 0 – TAI

Bits	Field	Type	Description
3	Block Update	RWR	Block Update. 0 = Update the 50% duty cycle clock mode registers as written 1 = Update the 50% duty cycle clock mode registers as a block When the 50% duty cycle clock mode is enabled (see TrigMode below), the following registers are used to configure that mode: TrigGenAmt (TAI offsets 0x02 & 0x03), TrigClkComp (TAI offset 0x04) and TrigClkCompSubPs (TAI offset 0x05). To compensate for PPM drift between this node's crystal and the Grand Master's crystal, these registers may need to be updated periodically. The same is true for the IRL clock which is configured using IRLClkGenAmt (TAI offset 0x06), IRLClkComp (TAI offset 0x07) and IRLClkCompSupPs (TAI offset 0x08). Setting this register bit to a one allows the updated values in these register groups (TAI offset 0x02 to 0x05 and TAI offset 0x06 to 0x08) to be presented to the hardware together at the same time and at a time when the hardware is not using these register values. This mode ensures smooth, glitch free, updates when the contents of more than one register needs to change during an update. The contents of the TRIG registers (TAI offsets 0x02 to 0x05) are presented to the hardware as a block whenever the TrigClkCompSubPs register is written to (at TAI offset 0x05). The contents of the IRL_Clk registers (TAI offsets 0x06 to 0x08) are presented to the hardware as a block whenever the IRLClkCompSubPs register is written to (at TAI offset 0x08). This means that the software doesn't need to write all of these registers during an update. Only the registers that are changing need to be written to (it is assumed that the sub pico second register will need to be adjusted for each update, so writing to this register triggers the update).
2	MultiPTP Sync	RWR	Multiple PTP device sync mode. Used in systems where multiple PTP enabled devices' need to synchronize their PTP Global Time counters (TAI offsets 0x0E & 0x0F). 0 = Normal Event Request mode 1 = Enable Multiple PTP device sync mode When this bit is cleared to zero, the EventRequest interface operates normally (i.e. an EventReq transfers the value of the PTP Global Time[31:0] register to the EventCapTime[31:0] register based on the setting of the EventCapOv register above). When this bit is set to a one an EventReq (see EventPhase above) transfers the value in TrigGenAmt[31:0] (TAI offsets 0x02 & 0x03) into the PTP Global Time[31:0] register (TAI offsets 0x0E & 0x0F). The EventCapTime[31:0] (TAI offsets 0x0A & 0x0B) is also updated at the same time with the previous value that the PTP Global Time[31:0] register contained prior to be updated.

Table 251: TAI Global Config, AVBPort = 0x1E
Offset: 0x00 or decimal 0 – TAI

Bits	Field	Type	Description
1	TrigMode	RWR	Trigger Mode. 0 = 50% duty cycle clock mode 1 = Pulse (one-shot) mode When this bit is cleared to zero the 50% duty cycle clock mode is enabled. In this mode the value specified in the TrigGenAmt is used as the period for generating a 50% duty cycle clock on the PTP_TRIG signal. Note that the minimum clock period that can be generated on the PTP_TRIG signal is 4 times the TSClkPer amount. The frequency of this clock can be adjusted in ps increments (see TrigClkComp, TAI offset 0x04) and it can be realigned to a specific time (see TrigLock bit above). The first leading edge of the 50% duty cycle clock will occur the first time the PTP Global Time (PTP TAI offsets 0x0E & 0x0F) equals the value in the non-zero TrigGenTime register (PTP TAI offsets 0x10 & 0x11) after TrigGenReq, below, is set to a one. This leading edge control occurs as long as the TrigGenTime register is non-zero. If it is zero the leading edge will occur when the TrigGenReq bit below is set to a one without regard to the PTP Global Time. The phase of the leading edge is controlled by the TrigPhase bit above. When this bit is set to a one, Pulse mode is enabled. This mode matches the PTP Global Timer (TAI offsets 0x0E & 0x0F) and the TrigGenAmt register (TAI ofsets 0x02 & 0x03) to generate a pulse at that time on the PTP_TRIG signal. The width of the pulse is specified by PulseWidth & PulseWitdhRange (TAI offset 0x05). Note: The minimum pulse width that can be generated is one TSClkPer amount (TAI offset 0x01) and the maximum pulse width is more than 30 million times the TSClkPer. The phase of the pulse is controlled by TrigPhase bit above.
0	TrigGen Req	RWR or SC	Trigger Generation Request/Enable. When this bit is set to a one, it enables a one-shot pulse or the 50% duty cycle clock generation on PTP_TRIG as previously configured by the TrigGenAmt, TrigMode, TrigClkComp (ps & Sub ps), PulseWidth, and TrigGenTime fields. If TrigMode (above) is set to a one (pulse mode) this bit will self clear after the pulse occurs (i.e., the trailing edge of the pulse as defined by the Pulse Width and Pulse Width Range registers (TAI offset 0x05). If TrigMode is cleared to a zero the 50% duty cycle clock will continue running as long as this bit is set to a one.

Table 252: TAI Global Config, AVBPort = 0x1E
Offset: 0x01 or decimal 1 – TAI

Bits	Field	Type	Description
15:0	TSClkPer	RWS 0x1F40	Time Stamping Clock Period in pico seconds. This field specifies the clock period for the time stamping clock supplied to the PTP hardware. When this device is using the 125 MHz internally generated clock for the PTP hardware, the value of this register must be 0x1F40, or 8000 decimal which indicates a clock period of 8000 ps or 8 ns (or 125 MHz). When this device's PTP hardware is clocked by an external clock (using PTP_EXTCLK – see PtpExtClk in TAI offset 0x08) this register must be set to the number of ps in that clock's period.

Table 253: TAI Global Config, AVBPort = 0x1E
Offset: 0x02 or decimal 2 – TAI

Bits	Field	Type	Description
15:0	TrigGen Amt [15:0]	RWR	Trigger Generation Amount bits 15 to 0 of a 32-bit register. This field specifies the PTP Time Application Interface trigger generation time amount. When TrigMode is set to one, the value specified in this field is compared with the PTP Global Timer (TAI offset 0x09 & 0x0A) and when it matches the first time, a pulse is generated on PTP_TRIG whose width is controlled by PulseWidth (TAI offset 0x05). In this mode there is an internal delay of three TSClkPer before the leading edge of the pulse will be seen on the PTP_TRIG output pin. When TrigMode is cleared to zero, the value in this field is used as a clock period in TSClkPer increments (TAI offset 0x01) to generate an output clock on the PTP_TRIG signal (see TrigPhase in TAI offset 0x00). In this mode the TrigClkComp amount (TAI offset 0x04) and TrigClkCompSubPs (TAI offset 0x05) gets accumulated once per TrigGenAmt cycle and when this accumulated value exceeds the value specified in TSClkPer, one TSClkPer amount gets added to or subtracted from the next trailing edge of PTP_TRIG clock output. **NOTE:** In 50% duty cycle clock mode the contents of this register can be presented to the hardware at the same time all the other 50% duty cycle registers are. See BlockUpdate in TAI offset 0x00. The upper 16-bits of this register are contained in the register below.

Table 254: TAI Global Config, AVBPort = 0x1E
Offset: 0x03 or decimal 3 – TAI

Bits	Field	Type	Description
15:0	TrigGen Amt [31:16]	RWR	Trigger Generation Amount bits 31:16 of a 32-bit register. This field specifies the PTP Time Application Interface trigger generation time amount. See the description above. The lower 16-bits of this register are contained in the register above.

Table 255: TAI Global Config, AVBPort = 0x1E
Offset: 0x04 or decimal 4 – TAI

Bits	Field	Type	Description
15	TrigComp Dir	RWR	Trig Clock Compensation Direction. When the accumulated TrigClkComp amount (below) exceeds the value in TSClkPer (TAI offset 0x01), one TSClkPer amount gets added to or subtracted from the next PTP_TRIG clock output. This bit determines which as follows: 0 = Add one TSClkPer to the next PTP_TRIG cycle 1 = Subtract one TSClkPer from the next PTP_TRIG cycle
14:0	TrigClk Comp	RWR	Trigger mode Clock Compensation Amount in pico seconds as an unsigned number. This field is used in 50% duty cycle clock mode only (when TrigMode is cleared to zero and TrigGenReq is set to one). This field specifies the remainder amount in ps for the clock that is being generated with a period specified by the TrigGenAmt (TAI offset 0x02 & 0x03). This field must be set as an absolute error number in ps (in other words it is a magnitude difference) regardless if the local clock is too fast or too slow compared to the reference clock. The direction of the clock compensation is configured in the CompDir bit above. In the 50% duty cycle clock mode this register gets accumulated once per TrigGenAmt cycle and when this accumulated value exceeds the value specified in TSClkPer (TAI offset 0x01), one TSClkPer amount gets added to or subtracted from the next PTP_TRIG clock output. This requires that the absolute value of TrigClkComp must not exceed the size of the TSClkPer. If it does, the TSClkPer needs to be adjusted in size (either up or down) until the remainder that remains is less that the TSClkPer and that value gets put into this register. **NOTE:** In 50% duty cycle clock mode the contents of this register can be presented to the hardware at the same time all the other 50% duty cycle registers are. See BlockUpdate in TAI offset 0x00.

Table 256: TAI Global Config, AVBPort = 0x1E
Offset: 0x05 or decimal 5 – TAI

Bits	Field	Type	Description
15:12	Pulse Width	RWS 0xF	Pulse width for PTP_TRIG. This pulse width is in units of TSClkPer (TAI offset 0x01). This specifies the width of the pulse that gets generated on PTP_TRIG (see TrigPhase in TAI offset 0x00) when the one shot pulse mode is selected (TrigMode is set to one and TrigGenReq is set to one). If the PTP_EXTCLK is not used (i.e., the internal clock is being used - see PtpExtClk in TAI offset 0x08) the TSClkPer, or Pulse Width unit is 8ns (assuming the Pulse Width Range, below, is 0x0). Note: Setting this register to 0x0 will cause unpredictable results.
11	Reserved	RES	Reserved for future use.
10:8	Pulse Width Range	RWR	Pulse Width Range for the PTP_TRIG signal. This register selects the units of time used to define the Pulse Width (above) as follows (each higher numbered selection generates units that are 8x larger than the previous lower numbered selection): 0x0 = 8 nSec units for a 125 MHz PTP clock or 1 x TSClkPer 0x1 = 64 nSec units for a 125 MHz PTP clock or 8 x TSClkPer 0x2 = 512 nSec units for a 125 MHz PTP clock or 64 x TSClkPer 0x3 = 4,096 nSec units for a 125 MHz PTP clock or 512 x TSClkPer 0x4 = 32.768 uSec units for a 125 MHz PTP clock or 4,096 x TSClkPer 0x5 = 262.144 uSec units for a 125 MHz PTP clock or 32,768 x TSClkPer 0x6 = 2.097 mSec units for a 125 MHz PTP clock or 262,144 x TSClkPer 0x7 = 16.777 mSec units for a 125 MHz PTP clock or 2,097,152 x TSClkPer The narrowest width is 8 nSec (assuming a 125 MHz PTP clock) or one TSClkPer (by setting this register to 0x0 and the Pulse Width register, above, to 0x1). The widest width is 251 mSec (assuming a 125 MHz PTP clock) or just a bit over ¼ second (by setting this register to 0x7 and the Pulse Width register to 0xF). The maximum number is 31,457,280 TSClkPer.

Table 256: TAI Global Config, AVBPort = 0x1E (Continued)
Offset: 0x05 or decimal 5 – TAI

Bits	Field	Type	Description
7:0	TrigClk Comp SubPs	RWR	Trigger mode Clock Compensation Amount in Sub Pico seconds as an unsigned number. This field is used in 50% duty cycle clock mode only (when TrigMode is cleared to zero and TrigGenReq is set to one). This field specifies the remainder amount in sub ps increments for the clock that is being generated with a period specified by the TrigGenAmt (TAI offset 0x02 & 0x03). This field must be set as an absolute error number in Sub ps (in other words it is a magnitude difference) regardless if the local clock is too fast or too slow compared to the reference clock. The direction of the clock compensation is configured in the CompDir bit above. Each unit in this register is 1/256th of a ps or approximately 4 femto seconds (actual number is 3.90625 femto seconds per unit). In the 50% duty cycle clock mode this register gets accumulated once per TrigGenAmt cycle and when this accumulated value exceeds on ps, one ps gets added to the accumulated Trig Clock Compensation (TAI offset 0x04). **NOTE:** In 50% duty cycle clock mode the contents of this register can be presented to the hardware at the same time all the other 50% duty cycle registers are. The writing to this register is used to transfer this data as a block. See BlockUpdate in TAI offset 0x00.

Table 257: TAI Global Config, AVBPort = 0x1E
Offset: 0x06 or decimal 6 – TAI

Bits	Field	Type	Description
15:0	IRLClk GenAmt [15:0]	RWS to 0x0186	Ingress Rate Limiter Clock Generation Amount bits 15 to 0. The value in this field is used as a clock period in TSClkPer increments (TAI offset 0x01) to generate a clock to the Ingress Rate Limiter (IRL) block's ITSM (Isochronious Time Slot Metering) mode (see Global 2 offsets 0x09 & 0x0A). In this mode the IRLClkComp amount (TAI offset 0x07) and IRLClkCompSubPs (TAI offset 0x08) gets accumulated once per IRLClkGenAmt cycle and when this accumulated value exceeds the value specified in TSClkPer, one TSClkPer amount gets added to or subtracted from the next trailing edge of IRL clock. **NOTE:** The contents of this register can be presented to the hardware at the same time all the other IRL_Clk registers are. See BlockUpdate in TAI offset 0x00. **NOTE:** The default value of this register (along with the IRLClkComp register below) generates a 3.125 uSec clock assuming the device's internal 125 MHz clock is being used.

Table 258: TAI Global Config, AVBPort = 0x1E
Offset: 0x07 or decimal 7 – TAI

Bits	Field	Type	Description
15	IRLComp Dir	RWR	IRL Clock Compensation Direction. When the accumulated IRLClkComp amount (below) exceeds the value in TSClkPer (TAI offset 0x01), one TSClkPer amount gets added to or subtracted from the next IRL clock. This bit determines which as follows: 0 = Add one TSClkPer to the next IRL clock cycle 1 = Subtract one TSClkPer from the next IRL clock cycle
14:0	IRLClk Comp	RWS to 0x1388	IRL Clock Compensation Amount in pico seconds as an unsigned number. This field specifies the remainder amount in ps for the clock that is being generated with a period specified by the IRLClkGenAmt (TAI offset 0x06). This field must be set as an absolute error number in ps (in other words it is a magnitude difference) regardless if the local clock is too fast or too slow compared to the reference clock. The direction of the clock compensation is configured in the CompDir bit above. This register gets accumulated once per TrigGenAmt cycle and when this accumulated value exceeds the value specified in TSClkPer (TAI offset 0x01), one TSClkPer amount gets added to or subtracted from the next IRL clock. This requires that the absolute value of IRLClkComp must not exceed the size of the TSClkPer. If it does, the TSClkPer needs to be adjusted in size (either up or down) until the remainder that remains is less that the TSClkPer and that value gets put into this register. **NOTE:** The contents of this register can be presented to the hardware at the same time all the other IRL_Clk registers are. See BlockUpdate in TAI offset 0x00.

Table 259: TAI Global Config, AVBPort = 0x1E
Offset: 0x08 or decimal 8 – TAI

Bits	Field	Type	Description
15:8	Reserved	RES	Reserved for future use.
7:0	IRLClk Comp SubPs	RWR	IRL Clock Compensation Amount in Sub Pico seconds as an unsigned number. This field specifies the remainder amount in sub ps increments for the clock that is being generated with a period specified by the IRLGenAmt (TAI offset 0x06). This field must be set as an absolute error number in Sub ps (in other words it is a magnitude difference) regardless if the local clock is too fast or too slow compared to the reference clock. The direction of the clock compensation is configured in the CompDir bit above. Each unit in this register is 1/256th of a ps or approximately 4 femto seconds (actual number is 3.90625 femto seconds per unit). This register gets accumulated once per IRLClkGenAmt cycle and when this accumulated value exceeds on ps, one ps gets added to the accumulated IRL Clock Compensation (TAI offset 0x07). **NOTE:** The contents of this register can be presented to the hardware at the same time all the other IRL_Clk registers are. The writing to this register is used to transfer this data as a block. See BlockUpdate in TAI offset 0x00.

Table 260: TAI Global Status, AVBPort = 0x1E
Offset: 0x09 or decimal 9 – TAI

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	Capture Trig	RWR	Capture Trig. 0 = Capture PTP_EVREQ pin events 1 = Capture PTP_TRIG internal events When this bit is cleared to a zero the Event Capture register looks at events on the PTP_EVREQ pin. When this bit is set to a one the Event Capture register looks at events from the waveform generated by PTP_TRIG. This allows observing the rising or falling edge of the PTP_TRIG (the EventPhase register is still active, PTP TAI offset 0x00) without the need of using pins. This is used to ensure the edges have not drifted over time so they can be re-aligned if needed.
13:10	Reserved	RES	Reserved for future use.
9	EventCap Err	RWR	Event Capture Error. This bit gets set by the hardware logic when an EventReq has occurred (see EventPhase in TAI offset 0x00) where the EventCapValid bit, below, is already set to a one and the EventCapOv bit (TAI offset 0x00) is cleared to a zero. This condition could happen if the EventReqs are occurring faster than the local CPU can process them (and clear the EventCapValid bit before the next EventReq). Some number of missed EventReq can be seen in the EventCapCtr, below, if its enabled.
8	EventCap Valid	RWR	Event Capture Valid. This bit is set to a one whenever the EventCap (Event Capture – TAI offsets 0x0A & 0x0B) register contains the time of a captured event. Software needs to clear this bit to a zero to enable the EventCap Register to be able to acquire a subsequent event if the EventCapOv (Event Capture Override – TAI offset 0x00) is not enabled. Clearing this bit to a zero also clears the EventInt (Event Capture Interrupt – PTP Global offset 0x08).
7:0	EventCap Ctr	RWR	Event Capture Counter. This field is incremented once by each EventReq (see EventPhase in PTP TAI offset 0x00) as long as EventCtrStart (PTP TAI offset 0x00) is set to one. This counter wraps around and can be cleared by writing zeros to it.

Table 261: TAI Global Status, AVBPort = 0x1E
Offset: 0x0A or decimal 10 – TAI

Bits	Field	Type	Description
15:0	EventCap Register [15:0]	RWR	Event Capture Register bits 15 to 0 of a 32-bit register. This register captures the value of the PTP Global Timer (TAI offsets 0x0E & 0x0F) when an EventReq (see EventPhase in TAI offset 0x00) has occurred. If the EventCapOv (TAI offset 0x00) is set to a one, then this register indicates the time captured for the last event. When EventCapErr is set to a one, the contents in this register no longer represent the time of the first event. **NOTE:** The maximum jitter for the EventCapRegister time amount with respect to the EventReq on a GPIO pin is one TSClkPer amount. **NOTE:** The minimum EventReq GPIO input signal high or low width has to be equal to or greater than 1.5 times the TSClkPer amount. **NOTE:** In order for hardware to capture the EventReq on the GPIO input signal, the minimum gap between two consecutive events has to be 150 ns plus 5 times the TSClkPer amount. The upper 16-bits of this register are contained in the register below.

Table 262: TAI Global Status, AVBPort = 0x1E
Offset: 0x0B or decimal 11 – TAI

Bits	Field	Type	Description
15:0	EventCap Register [31:16]	RWR	Event Capture Register bits 31 to 16 of a 32-bit register. This register captures the value of the PTP Global Timer (TAI offsets 0x0E & 0x0F) when an EventReq (see EventPhase in TAI offset 0x00) has occurred. See the description above. The lower 16-bits of this register are contained in the register above.

Table 263: TAI Global Status, AVBPort = 0x1E
Offset: 0x0E or decimal 14 – TAI

Bits	Field	Type	Description
15:0	PTPGlobalTime [15:0]	RO	Precise Time Protocol Global Timer bits 15 to 0 of a 32-bit register. This indicates the global timer value that is running off of the free running switch core clock. This counter wraps around in hardware. To support synchronization of PTP Global Time between multiple devices in a system, this register gets updated with the value specified in TrigGenAmt when MultiPTPSync is set to a one (TAI offset 0x00) and an EventReq occurs (see EventPhase in TAI offset 0x00). The upper 16-bits of this register are contained in the register below.

Table 264: TAI Global Status, AVBPort = 0x1E
Offset: 0x0F or decimal 15 – TAI

Bits	Field	Type	Description
15:0	PTPGlobalTime [31:16]	RO	Precise Time Protocol Global Timer bits 31 to 16 of a 32-bit register. This indicates the global timer value that is running off of the free running switch core clock. This counter wraps around in hardware. See the description above. The lower 16-bits of this register are contained in the register above.

Table 265: TAI Global Config, AVBPort = 0x1E
Offset: 0x10 or decimal 16 – TAI

Bits	Field	Type	Description
15:0	TrigGen Time [15:0]	RWR	Trigger Generation Time bits 15 to 0 of a 32-bit register. This field specifies the PTP Global Time (TAI offsets 0x0E & 0x0F) where the 1st leading edge of PTP_TRIG will occur (with a three TSClkPer latency) when PTP Trig is in the continuous square wave mode (i.e, when TrigMode is 0x0, offset 0x00 above) as long as this register's value is non-zero. If its value is zero, the 1st leading edge of PTP_TRIG will occur when TrigGenReg is set to a one (TAI offset 0x00). This register is also used to for re-locking the leading edge of the square wave (see TrigLock in TAI offset 0x00). The upper 16-bits of this register are contained in the register below.

Table 266: TAI Global Config, AVBPort = 0x1E
Offset: 0x11 or decimal 17 – TAI

Bits	Field	Type	Description
15:0	TrigGen Time [31:16]	RWR	Trigger Generation Time bits 31 to 16 of a 32-bit register. See the description above. The lower 16-bits of this register are contained in the register above.

Table 267: TAI Global Config, AVBPort = 0x1E
Offset: 0x12 or decimal 18 – TAI

Bits	Field	Type	Description
15:5	Reserved	RES	Reserved for future use
4	LockCorr Valid	RO	Trig Lock Correction Valid. 0 = Trig Lock Correction is not valid or did not occur 1 = Trig Lock Correction is valid & did occur When a Trigger Lock is enabled (TAI offset 0x00) this bit is cleared to zero. When the Trigger Lock completes this bit will be set to a one if and only if a Trigger Lock occurred for PTP_TRIG. In this case the Lock Correction value below will show the amount of adjustment that was made (if any).
3:0	Lock Correction	RO	Trig Lock Correction amount. When the TrigLock bit is set to a one (TAI offset 0x00) enabling a potential clock adjustment, these bits are cleared to zero. When the TrigLock bit is cleared to zero (indicating that the requested clock adjustment is now past in time) these bits will reflect the magnitude and direction that was applied to the PTP_TRIG leading edge of the generated clock. A value of zero means no adjustment was necessary. If bit 3 is a one then the leading edge of the clock was moved n number of clocks earlier in time where n is shown in bits 2:0. If bit 3 is a zero then the leading edge of the clock was moved n number of clocks later in time where n is shown in bits 2:0.

Table 268: TAI Global Config, AVBPort = 0x1E
Offset: 0x1E or decimal 8 – TAI

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	PtpExtClk	RWR	PTP External Clock select. 0 = Use internal clock for the PTP core 1 = Use external clock for the PTP core When this bit is cleared to a zero the PTP core gets its clock from an internal 125 MHz clock based on the device's XTAL_IN input. When this bits is set to a one the PTP core gets its clock from the device's PTP_EXTCLK pin. **NOTE:** Do not select the PTP_EXTCLK pin unless the pin has a clock and one of the GPIO pins is configured to be the PTP_EXTCLK (Global 2 offset 0x1A indexes 0x68 to 0x6F).
13:11	Reserved	RES	Reserved for future use.
10:8	SecRec ClkSel	RWS to 0x7	Synchronous Ethernet Secondary Recovered Clock Select & Qbv 1 Debug. This field indicates the internal PHY number whose recovered clock will be presented on the GPIO pin which is configured to select RX_CLK1 or it selects Qbv 1 for debug. The reset value selects Qbv 1 for debug which results in no clock for RX_CLK1 if Qbv debug is not being used. The RX_CLK1 frequency is 25 MHz when SecRecClkSel points to a physical PHY or SERDES (2 for PHY 2, 3 for PHY 3, etc.). Selecting a port where there is no PHY (or SERDES) will connect the RX_CLK1 signal to the internal 25 MHz XTAL clock. When SecRecClkSel connects to Qbv 1 the egress port that is connected to RX_CLK1 for Qbv debug is determined by PortSel1 in Qbv Global offset 0x08.
7:3	Reserved	RES	Reserved for future use.
2:0	PriRec ClkSel	RWS to 0x7	Synchronous Ethernet Primary Recovered Clock Select & Qbv 0 Debug. This field indicates the internal PHY number whose recovered clock will be presented on the GPIO pin which is configured to select RX_CLK0 or it selects Qbv 0 for debug. The reset value selects Qbv 0 for debug which results in no clock for RX_CLK0 if Qbv debug is not being used. The RX_CLK0 frequency is 25 MHz when PriRecClkSel points to a physical PHY or SERDES (2 for PHY 2, 3 for PHY 3, etc). Selecting a port where there is no PHY (or SERDES) will connect the RX_CLK0 signal to the internal 25 MHz XTAL clock. When PriRecClkSel connects to Qbv 0 the egress port that is connected to RX_CLK0 for Qbv debug is determined by PortSel0 in Qbv Global offset 0x08.

1.5.3 AVB Policy Registers

This section describes the AVB Policy registers. The following are the register bits used for configuration and status information to and from the software / CPU sub-system for Precise Time Protocol logic for audio-video bridging applications. These registers accessed using the Global 2 register s AVB Command and AVB Data registers (Offset 0x16 and Offset 0x17).

Table 269: AVB Policy Port Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x1 & AVBPort = 6:0)

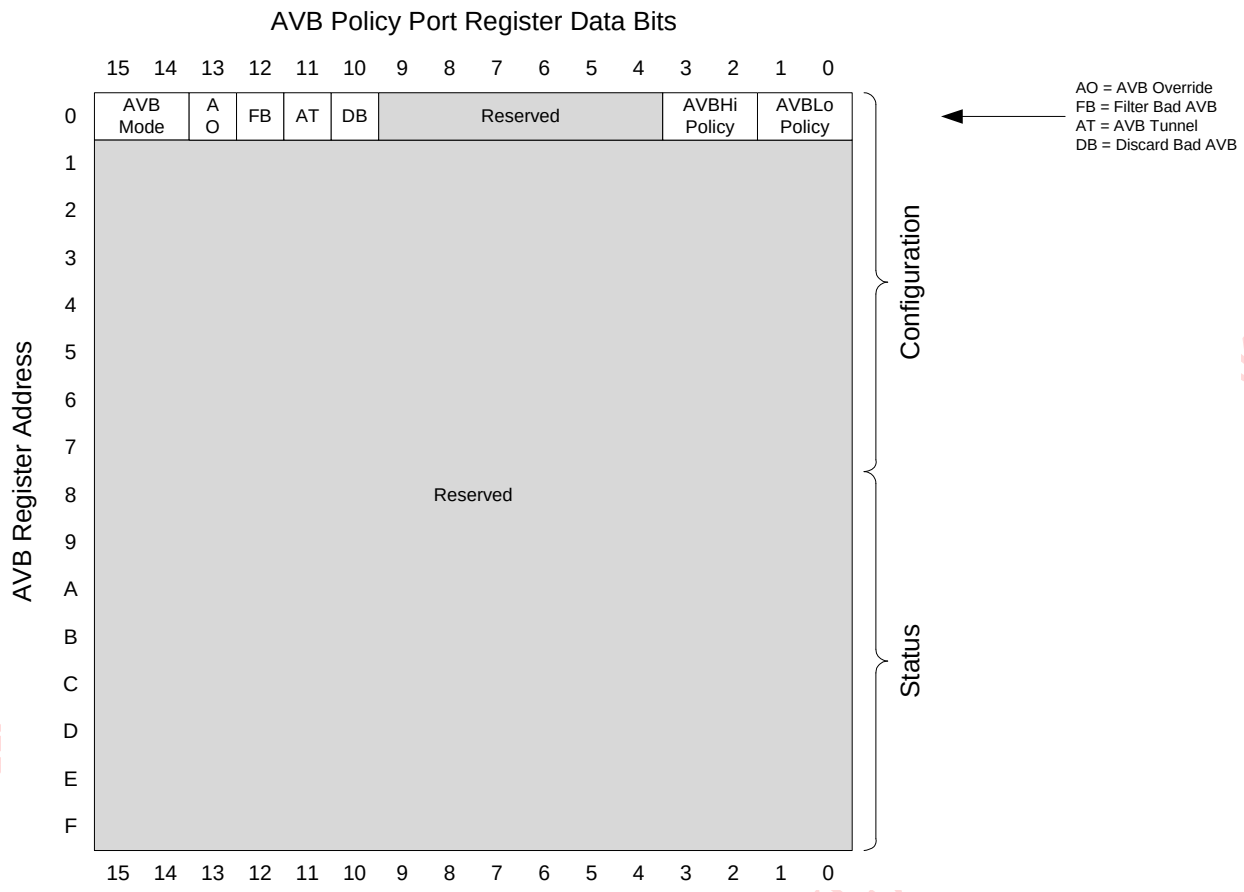


Table 270: AVB Policy Port Register
Offset: 0x0

Bits	Field	Type	Description
15:14	AvbMode	RWR	Port's AVB Mode. These bits are used to select the AVB mode on the port as follows: 0x0 = Legacy port mode. All frames entering this port are considered Legacy. unless they are overridden by the frame's DA[1] in which case they are considered AVB. 0x1 = Standard AVB port mode. Any tagged frame (Provider or 802.1Q tagged) that ends up with[2] an AVB frame priority[3] is considered AVB. All other frames are considered Legacy. 0x2 = Enhanced AVB port mode. Any frame that ends up with an AVB frame priority whose DA is contained in the ATU with an AVB Entry State is considered AVB. All other frames are considered Legacy. Frames that end up with an AVB frame priority but whose DA is not in ATU with an AVB Entry State are considered Bad AVB frames (which can be filtered – see Filter Bad Avb below). 0x3 = Secure AVB port mode. Any frame that ends up with an AVB frame priority whose DA is contained in the ATU with an AVB Entry State and whose DPV has this source port's bit set to a one, is considered AVB. All other frames are considered Legacy. Frames that end up with an AVB frame priority but whose DA is not in ATU with an AVB Entry State or whose DPV does not have this source port's bit set to a one, are considered Bad AVB frames (which can be filtered – see Filter Bad Avb below). AVB frames are allowed to use the AVB queues on other AVB egress ports (those ports where AvbMode <> 0x0 – see AvbOverride below). Legacy frames cannot use AVB queues.
13	Avb Override	RWR	AVB Override. When this bit is cleared to a zero, normal frame processing occurs. When this bit is set to a one the egress portion of this port is considered AVB even if the ingress portion is not (because AvbMode, above, is set to 0x0). This allows this port's egress to support AVB queues and flows even though the port is a Legacy port. This can be useful if an AVB flow is allowed to egress the AVB cloud.
12	FilterBad Avb	RWR	Filter Bad AVB frames. When this bit is cleared to a zero, normal frame processing occurs. When this bit is set to a one, frames that are considered Bad AVB frames (see AvbModes 2 & 3 above) are filtered using the method determined by the DiscardBadAvb bit below. This can be useful to prevent AVB steams from flowing before the path is completely set up.

Table 270: AVB Policy Port Register (Continued)
Offset: 0x0

Bits	Field	Type	Description
11	AvbTunnel	RWR	AVB Tunnel. When this bit is cleared to a zero normal frame processing occurs. 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 that is considered AVB by DA. This includes unicast as well as multicast frames. A frame is considered AVB by DA if its DA is in the ATU with an AVB Entry State with priority override where the overridden priority equals the Hi or Lo AVB frame priorities (AVB Policy Global offset 0x00) and where the port's DA Priority Override bits are 0x1 (bits 15:14 of this register). Note: Do not set this bit to a 0x1 if the port's VLANTunnel bit (Port offset Register 0x04) is set to a 0x1.
10	Discard BadAvb	RWR	Discard Bad AVB frames. When the FilterBadAvb bit, above, is set to a one this bit determines the type of filtering that will occur on frames that are considered Bad AVB frames (see AvbModes 2 & 3 above). When this bit is cleared to a zero, Bad AVB frames are prevented from egressing out AVB ports only. In this mode Bad AVB frames are still allowed to egress non-AVB ports, however. This keeps the Bad AVB frames out of the AVB queues but allows them to egress non-AVB ports which may be best for the network. When this bit is set to a one, Bad AVB frames are prevented from egressing out all ports. **NOTE:** An AVB egress port is one whose AVB Mode is <> 0x0 or whose AVB Override bit is set to a one (see bits above).
9:4	Reserved	RES	Reserved for future use.

Table 270: AVB Policy Port Register (Continued)
Offset: 0x0

Bits	Field	Type	Description
3:2	Avb Hi Policy	RWR	Avb Hi 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 AVB Hi Policy Mapping can occur on this port for AVB Hi frames. AVB Hi Policy Mapping occurs when the DA of a frame is contained in the ATU address database (Global offset 0x0C) with an Entry State that indicates AVB with priority override where the overridden priority equals the Hi AVB frame priority (AVB Policy Global offset 0x00) and when the port's DA Priority Override bits are 0x1 (bits 15:14 of this register). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 0x3 = Reserved 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	Avb Lo Policy	RWR	Avb Lo 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 AVB Lo Policy Mapping can occur on this port for AVB Lo frames. AVB Lo Policy Mapping occurs when the DA of a frame is contained in the ATU address database (Global offset 0x0C) with an Entry State that indicates AVB with priority override where the overridden priority equals the Lo AVB frame priority (AVB Policy Global offset 0x00) and when the port's DA Priority Override bits are 0x1 (bits 15:14 of this register). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows: 0x0 = Normal frame switching 0x1 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 0x2 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 0x3 = Reserved 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. An AVB DA override requires the frame's DA to be in the ATU with an AVB Entry State with priority override and an AVB Hi or AVB Lo FPri in the ATU entry and the port's DA Priority Override mode (Port offset 0x0D) is set to 0x1.
2. Frame priority, or FPri, can be modified by many mechanisms inside the switch.
3. An AVB frame priority is an FPri that is equal to AvbHiFPri or AvbLoFPri (AVB Policy Global offset 0x00).

Table 271: AVB Policy Global Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x1 & AVBPort = 0x1F)

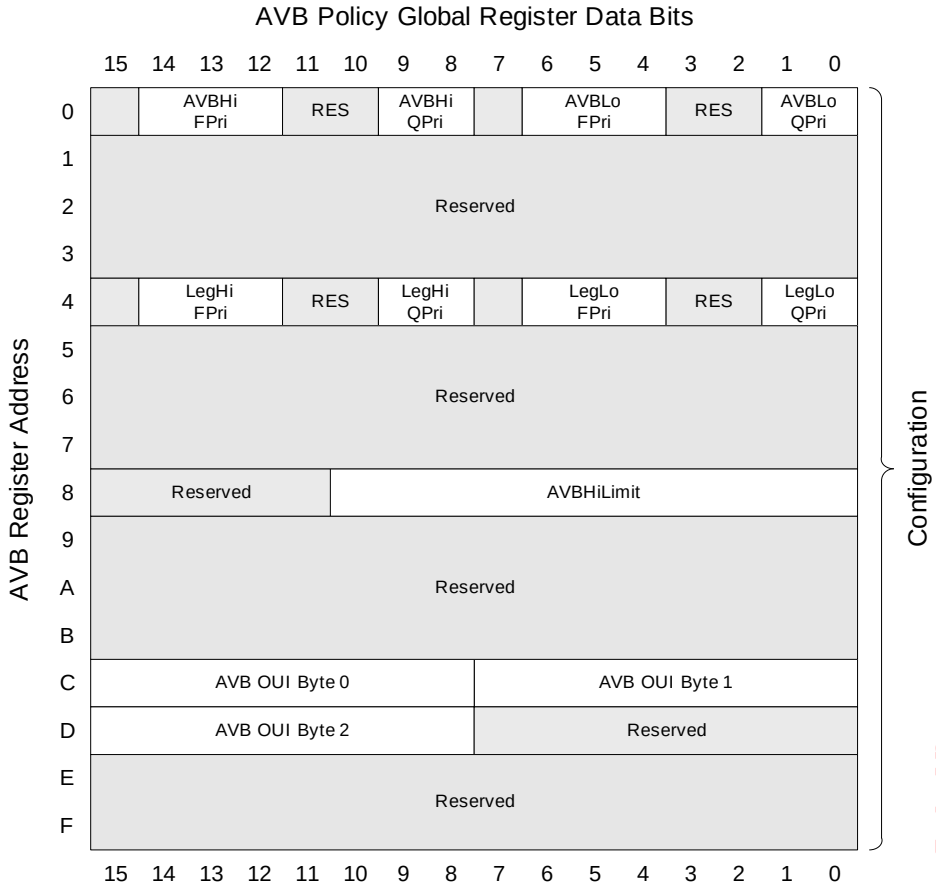


Table 272: AVB Policy Global AVB Register, AVBPort = 0xF
Offset: 0x0 or decimal 0

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:12	AvbHiFPri	RWS 0x5	AVB Hi Frame Priority. Used to define the frame priority used on the low latency, hi priority AVB flows. When a frame is assigned this FPri value in ingress it will be mapped into the AvbHiQPri egress queue (below) on AVB egress ports if this flow is determined to be AVB (see AvbMode, AVB Port offset 0x0). On AVB egress ports, any non-AVB frame with this FPri will be re-marked (assuming the frame is egressing tagged) with the LegacyHiFPri value (AVB Global offset 0x4).
11:10	Reserved	RES	Reserved for future use.
9:8	AvbHiQPri	RWS 0x3	AVB Hi Queue Priority. Used to define the queue in the device used for low latency, hi priority AVB flows. Any non-AVB frame (see AvbMode, AVB Port offset 0x0) with a QPri equal to this value will be mapped into the LegacyHiQPri queue (AVB Global offset 0x4) on AVB ports (see AvbMode and AvbOverride in AVB Port offset 0x0). The only exception to this is frames that get their QPriAvb assigned by the Priority Override Table (Global 2 offset 0x0F). **NOTE:** AVB class A or isochronous high traffic can only be mapped to queue numbers 3 and/or 2.
7	Reserved	RES	Reserved for future use.
6:4	AvbLoFPri	RWS 0x4	AVB Lo Frame Priority. Used to define the frame priority used on the higher latency, lo priority AVB flows. When a frame is assigned this FPri value in ingress it will be mapped into the AvbLoQPri egress queue (below) on AVB egress ports if this flow is determined to be AVB (see AvbMode, AVB Port offset 0x0). On AVB egress ports, any non-AVB frame with this FPri will be re-marked (assuming the frame is egressing tagged) with the LegacyLoFPri value (AVB Global offset 0x4).
3:2	Reserved	RES	Reserved for future use.
1:0	AvbLoQPri	RWS 0x2	AVB Lo Queue Priority. Used to define the queue in the device used for higher latency, lo priority AVB flows. Any non-AVB frame (see AvbMode, AVB Port offset 0x0) with a QPri equal to this value will be mapped into the LegacyLoQPri queue (AVB Global offset 0x4) on AVB ports (see AvbMode and AvbOverride in AVB Port offset 0x0). The only exception to this is frames that get their QPriAvb assigned by the Priority Override Table (Global 2 offset 0x0F). **NOTE:** AVB class B or isochronous low traffic can only be mapped to queue numbers 2 and/or 1.

Table 273: AVB Policy Global Legacy Register, AVBPort = 0xF
Offset: 0x4 or decimal 4

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:12	Legacy HiFPri	RWS 0x3	Legacy Hi Frame Priority. Used to remark frame priorities on non-AVB frames (see AvbMode, AVB Port offset 0x0) that egress AVB ports when these frames ended up with an FPri equal to AvbHiFPri (AVB Global offset 0x0).
11:10	Reserved	RES	Reserved for future use.
9:8	Legacy HiQPri	RWS 0x1	Legacy Hi Queue Priority. Used to define the queue in the device that non-AVB flows (see AvbMode in AVB Port offset 0x0) are mapped into on AVB ports (see AvbMode and AvbOverride in AVB Port offset 0x0) when the QPri assigned to the frame is equal to AvbHiQPri. The only exception to this is frames that get their QPriAvb assigned by the Priority Override Table (Global 2 offset 0x0F).
7	Reserved	RES	Reserved for future use.
6:4	Legacy LoFPri	RWS 0x2	Legacy Lo Frame Priority. Used to remark frame priorities on non-AVB frames (see AvbMode, AVB Port offset 0x0) that egress AVB ports when these frames ended up with an FPri equal to AvbLoFPri (AVB Global offset 0x0).
3:2	Reserved	RES	Reserved for future use.
1:0	Legacy LoQPri	RWS 0x1	Legacy Lo Queue Priority. Used to define the queue in the device that non-AVB flows (see AvbMode in AVB Port offset 0x0) are mapped into on AVB ports (see AvbMode and AvbOverride in AVB Port offset 0x0) when the QPri assigned to the frame is equal to AvbLoQPri. The only exception to this is frames that get their QPriAvb assigned by the Priority Override Table (Global 2 offset 0x0F).

Table 274: AVB Policy Global Limit Register, AVBPort = 0xF
Offset: 0x8 or decimal 8

Bits	Field	Type	Description
15:11	Reserved	RES	Reserved for future use
10:0	AVBHiLimit	RWR	AVB Hi Frame Limit. When these bits are zero normal frame processing occurs. When these bits are non-zero they are used to define the maximum frame size allowed for AVB Hi or Class A frames (see AvbMode in AVB Port offset 0x0) that can be placed into the AvbHiQPri queue (AVB Global offset 0x0). Frames that are over this size limit are filtered.

Table 275: AVB Policy Global OUI Register Byte 0 & 1, AVBPort = 0xF
Offset: 0xC or decimal 12

Bits	Field	Type	Description
15:8	AvbOui Byte0	RWR	AVB OUI Limit Filter Byte 0. When all three of the AvbOui Bytes are zero normal frame processing occurs. When any of the three AvbOui Bytes are non-zero, all frames that have a Destination Address (DA) whose 1st three bytes of the DA (the OUI) match these three AvbOui Byte registers must be good AVB frames (see AvbMode in AVB Port offset 0x0), otherwise the frames will be filtered. This prevents non-AVB frames from using this OUI range of MAC addresses.
7:0	AvbOut Byte1	RWR	AVB OUI Limit Filter Byte 1. See AvbOuiByte0 above.

Table 276: AVB Policy Global OUI Register Byte 2, AVBPort = 0xF
Offset: 0xD or decimal 13

Bits	Field	Type	Description
15:8	AvbOui Byte2	RWR	AVB OUI Limit Filter Byte 2. See AvbOuiByte0 in the register above.
7:0	Reserved	RES	Reserved for future use.

1.5.4 Qav Registers

This section covers the registers that are part of the AVB command (Global Register 2, Offset 0x16) address space in general and more specifically it covers the Qav registers. These registers accessed using the Global 2 register s AVB Command and AVB Data registers (Offset 0x16 and Offset 0x17).

Table 277: Qav Port Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x2 & AVBPort = 6:0)

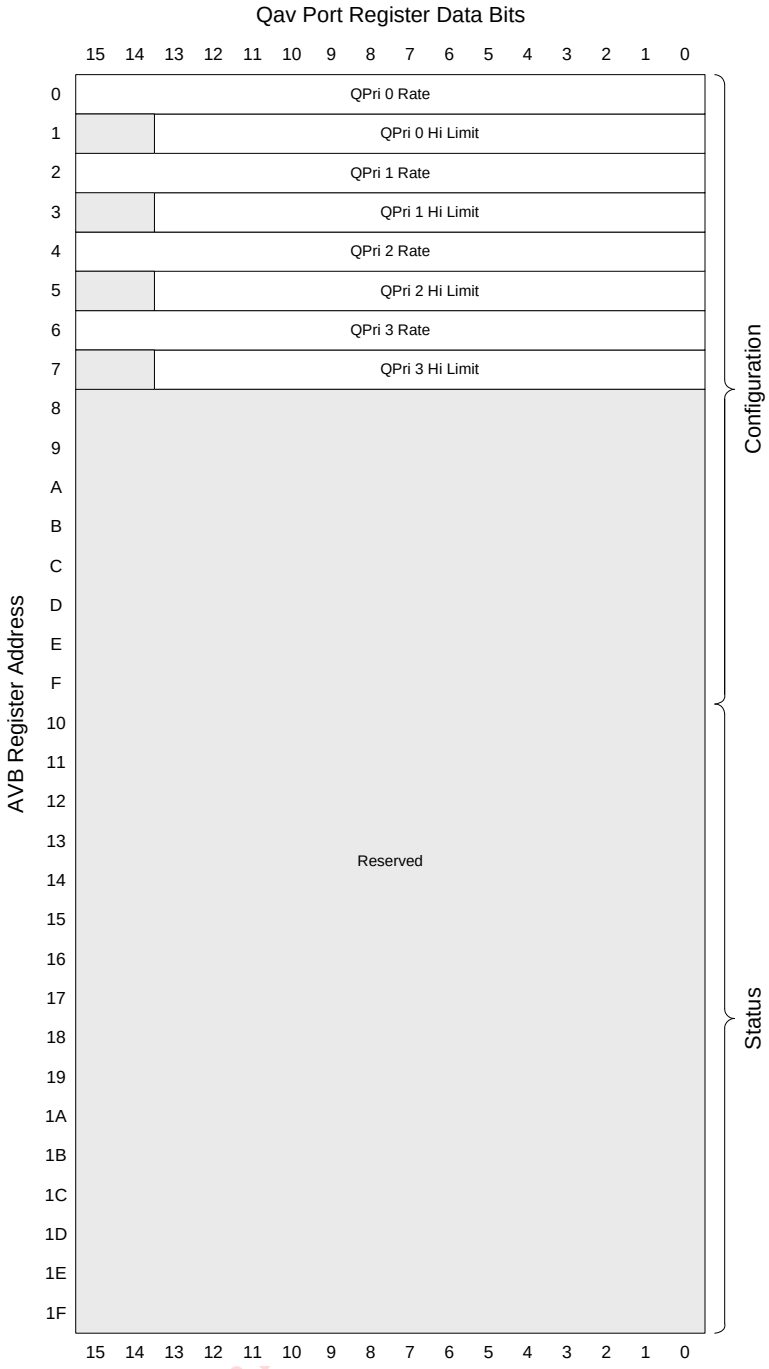


Table 278: Qav Port Config Register
Offset: 0x00 or decimal 0

Bits	Field	Type	Description
15:0	QPri 0 Rate	RWR	Priority Queue 0 Rate. A value of 0x0000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as: **QPri 0 Rate = Desired Rate / 64k bits per sec** For example, for a Desired Rate of 128 Kbps set this register to a value of 0x0002 (128 Kbps/ 64 Kbps = 2). **NOTE:** The minimum supported rate is 64 Kbps and the rate increments supported are 64 Kbps with a maximum supported rate of 4000 Mbps (i.e., the speed of the port).

Table 279: Qav Port Config Register
Offset: 0x01 or decimal 1

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:0	QPri0 HiLimit	RWS 0x3FFF	Priority Queue 0 Hi Limit. This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for this queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 16,384 bytes is enabled on this queue.

Table 280: Qav Port Config Register
Offset: 0x02 or decimal 2

Bits	Field	Type	Description
15:0	QPri1 Rate	RWR	Priority Queue 1 Rate. A value of 0x0000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as: **QPri 1 Rate = Desired Rate / 64k bits per sec** For example, for a Desired Rate of 128 Kbps set this register to a value of 0x0002 (128 Kbps/ 64Kbps = 2). **NOTE:** The minimum supported rate is 64 Kbps and the rate increments supported are 64 Kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).

Table 281: Qav Port Config Register
Offset: 0x03 or decimal 3

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:0	QPri1 HiLimit	RWS 0x3FFF	Priority Queue 1 Hi Limit. This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for this queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 16,384 bytes is enabled on this queue.

Table 282: Qav Port Config Register
Offset: 0x04 or decimal 4

Bits	Field	Type	Description
15:0	QPri 2 Rate	RWR	Priority Queue 2 Rate. A value of 0x0000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as: **QPri 2 Rate = Desired Rate / 64k bits per sec** For example, for a Desired Rate of 128 Kbps set this register to a value of 0x0002 (128 Kbps/ 64 Kbps = 2). **NOTE:** The minimum supported rate is 64 Kbps and the rate increments supported are 64 Kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).

Table 283: Qav Port Config Register
Offset: 0x05 or decimal 5

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:0	QPri2 HiLimit	RWS 0x3FFF	Priority Queue 2 Hi Limit. This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for the queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 16,384 bytes is enabled on this queue.

Table 284: Qav Port Config Register
Offset: 0x06 or decimal 6

Bits	Field	Type	Description
15:0	QPri3 Rate	RWR	Priority Queue 3 Rate. A value of 0x0000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as: **QPri 3 Rate = Desired Rate / 64k bits per sec** For example, for a Desired Rate of 128 Kbps set this register to a value of 0x0002 (128 Kbps/ 64Kbps = 2). **NOTE:** The minimum supported rate is 64 Kbps and the rate increments supported are 64 Kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).

Table 285: Qav Port Config Register
Offset: 0x07 or decimal 7

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:0	QPri3 HiLimit	RWS 0x3FFF	Priority Queue 3 Hi Limit. This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for the queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 16,384 bytes is enabled on this queue.

Table 286: Qav Global Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x2 & AVBPort = 0xF)

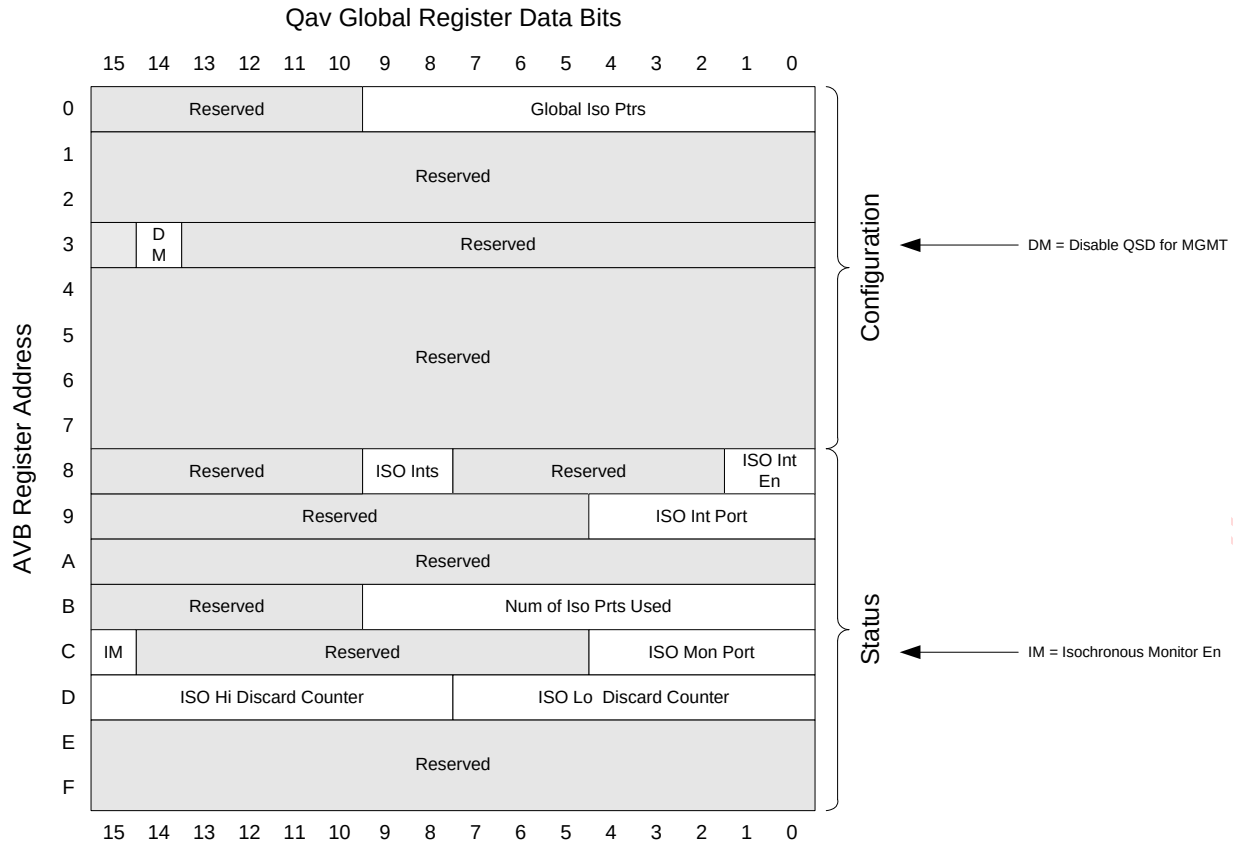


Table 287: Qav Global Config Register, AVBPort = 0xF
Offset: 0x00 or decimal 0

Bits	Field	Type	Description
15:10	Reserved	RES	Reserved for future use.
9:0	Global IsoPtrs	RWR	Global Isochronous Queue Pointer Threshold This field indicates the total number of isochronous pointers that are reserved for isochronous streams. The value is expected to be computed in SRP software and programmed into hardware based on the total aggregate isochronous streams configured to go through this device.

Table 288: Qav Global Config Register, AVBPort = 0xF
Offset: 0x03 or decimal 3

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	DisQSD forMGMT	RWR	Disable Queue Scheduler Delays for Management frames. This bit is operational when the MGMT frames are shared with any of the isochronous queues. When 0x1, it indicates to the Queue Controller to disable applying Queue Scheduler Delays and the corresponding rate regulator does not account for MGMT frames through this queue. When 0x0, the MGMT frames follow similar rate regulation and delay regulation envelope as specified for the isochronous queue that the MGMT frames are sharing with.
13:0	Reserved	RES	Reserved for future use.

Table 289: Qav Global Status Register, AVBPort = 0xF
Offset: 0x08 or decimal 8

Bits	Field	Type	Description
15:10	Reserved	RES	Reserved for future use.
9	Iso Dis Int	ROC	Isochronous packet discard Interrupt If the Queue controller had to discard an isochronous packet due to congestion reasons then this bit will get set. This indicates to the CPU that the configured SRP streams are not well behaved leading to congestion in Queue Controller. This field corresponds to the Iso Int Port information. When set, this bit gets cleared upon a read operation from CPU.
8	IsoLimit Ex Int	ROC	Isochronous Packet memory limit exceeded Interrupt. In order to guarantee that the isochronous streams always get packet memory pointers, GlobalIsoPtrs (Qav Global configuration data structure) is a threshold configured by SRP software layer based on the aggregate resources needed for the isochronous streams. This threshold will ensure that asynchronous streams don't end up occupying packet memory pointers allocated for the isochronous streams. But the isochronous streams are not prohibited from dipping into asynchronous memory pointers, even though this is expected to happen due to network mis-configuration. This interrupt bit is set by hardware when the Queue Controller exceeds the Isochronous GlobalIsoPtrs limit to accommodate an isochronous packet. When set, this bit gets cleared upon a read operation from CPU.
7:2	Reserved	RES	Reserved for future use.
1	IsoDisIntEn	RWR	Iso Discard Interrupt Enable. This bit must be set to a one to allow the Iso Discard interrupt to drive the device's INTn pin low (assuming the AVB Interrupts are unmasked in Switch Global Control, Global offset 0x04).
0	IsoLimitEx IntEn	RWR	Iso Packet Memory Exceeded Interrupt Enable. This bit must be set to a one to allow the Iso Packet Memory Exceeded interrupt to drive the device's INTn pin low (assuming the AVB Interrupts are unmasked in Switch Global Control, Global offset 0x04).

Table 290: Qav Global Status Register, AVBPort = 0xF
Offset: 0x09 or decimal 9

Bits	Field	Type	Description
15:4	Reserved	RES	Reserved for future use.
3:0	IsoIntPort	ROC	Isochronous interrupt port. This field indicates the port number for IsoDisInt or IsoLimitExInt bits. Only one such interrupt condition can be detected by hardware at one time. Once an interrupt bit has been set along with the IsoIntPort, the software would have to come and clear the bits before hardware records another interrupt event. **NOTE:** This field is valid for IsoDisInt interrupt condition only, i.e., for the IsoLimitExInt interrupt condition this field will be set to 0xF.

Table 291: Qav Global Status Register, AVBPort = 0xF
Offset: 0x0B or decimal 11

Bits	Field	Type	Description
15:0	Reserved	RES	Reserved for future use.
9:0	Numof IsoPtrs Used	RO	Number of Isochronous pointers used. This field indicates the queue controller status of number of iso pointers that are currently being used across various ports.

Table 292: Qav Global Status Register, AVBPort = 0xF
Offset: 0x0C or decimal 12

Bits	Field	Type	Description
15	Iso Mon En	RWR	Isochronous monitor enable When set to a one, this bit enables the statistics gathering capabilities stated in PTP Global Status Registers Offset 0x0D. Once enabled, the software is expected to program the IsoMonPort (below) indicating which port of the device does the software wants to monitor. Upon setting this bit, the hardware collects IsoHiDisCtr and IsoLoDisCtr values for the port indicated by IsoMonPort till this bit is set to a zero. When set to a zero, this bit disables the statistics gathering capabilities.
14:4	Reserved	RES	Reserved for future use.
3:0	Iso Mon Port	RWR	Isochronous monitoring port This field is updated by software along with Iso Mon En bit (above) and it indicates the port number that the software wants the hardware to start monitoring i.e., start updating IsoHiDisCt and, IsoLoDisCtr. The queue controller clears the above stats when IsoMonPort is changed.

Table 293: Qav Global Status Register, AVBPort = 0xF
Offset: 0x0D or decimal 13

Bits	Field	Type	Description
15:8	IsoHi DisCtr	RWR	Isochronous hi queue discard counter. This field is updated by hardware when instructed to do so by enabling the IsoMonEn bit in Qav Global Status Register Offset 0x0C. This is an upcounter of number of isochronous hi packets discarded by Queue Controller. This counter wraps around.
7:0	IsoLo DisCtr	RWR	Isochronous lo queue discard counter. This field is updated by hardware when instructed to do so by enabling the IsoMonEn bit in Qav Global Status Register Offset 0x0C. This is an upcounter of number of isochronous lo packets discarded by Queue Controller. This counter wraps around.

1.5.5 Qbv Registers – Time Aware AVB Gen 2/TSN Shaper

This section covers the registers that are part of the AVB command (Global Register 2, Offset 0x16) address space that are used to access the Qbv registers. These registers are accessed using the Global 2 AVB Command and AVB Data registers (Offset 0x16 and Offset 0x17) using an AVBBlock value of 0x3.

Table 294: Qbv Port Register bit Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x3 & AVBPort = n-1:0)

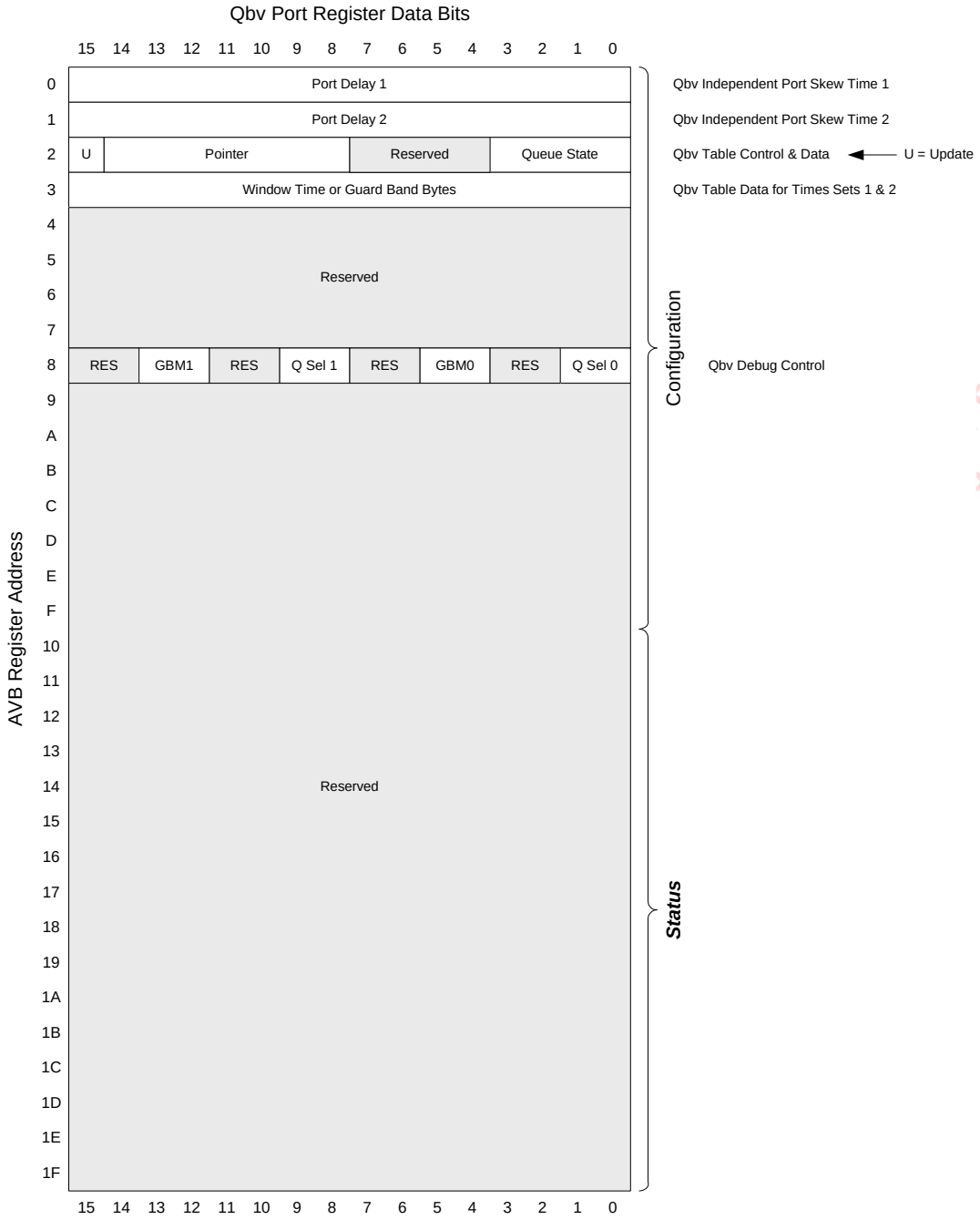


Table 295: Qbv Port Delay
Offset: 0x00 or decimal 0 – Qbv Port

Bits	Field	Type	Description
15:0	Port Delay 1	RWR	Port Delay Time 1. This is the delay from the Global t_0 (as defined by PTP Trig – see PTP TAI Global registers) to this Port's t_0 (where the Port's t_0 is the time where the port starts processing its Port Qbv Table Set 1). This register has a resolution of 64ns of delay per count so it has a range of 0 to 4.194 mSec. **NOTE:** If the 1st Qbv Window edge needs more accuracy than 64ns, set this Port Delay register to the maximum number of whole 64ns counts that can fit, then set the Qbv Table entry 0x00 to delay the rest of the time in 8ns resolution counts (using the current Queue State values so it acts like nothing more than a simple delay).

Table 296: Qbv Port Delay 2 Register
Offset: 0x01 or decimal 1 – Qbv Port

Bits	Field	Type	Description
15:0	Port Delay 2	RWR	Port Delay Time 2. This is the delay from the Global t_0 (as defined by PTP Trig – see PTP TAI Global registers) to this Port's t_0 (where the Port's t_0 is the time where the port starts processing its Port Qbv Table Set 2). This register has a resolution of 64ns of delay per count so it has a range of 0 to 4.194 mSec. **NOTE:** If the 1st Qbv Window edge needs more accuracy than 64ns, set this Port Delay register to the maximum number of whole 64ns counts that can fit, then set the Qbv Table entry 0x00 to delay the rest of the time in 8ns resolution counts (using the current Queue State values so it acts like nothing more than a simple delay).

Table 297: Qbv Port Table Control
Offset: 0x02 or decimal 2 – Qbv Port

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 of this register, along with bits 15:0 of the Qbv Table Data register at Qbv Port offset 0x03 will be loaded into the Qbv Table entry referenced by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	Pointer to the desired entry of Port Qbv Table. These bits select one of the possible Port Qbv Table entries for both read and write operations. A write operation occurs if the Update bit is a one (the Queue State bits in this register can be written at the same time by writing this register in a single operation – but the Guard Band and Time bits should be written prior to the writing of this register). Otherwise a read of the current Pointer occurs and the data found there is placed in the Queue State bits (in this register) & the Guard Band and Time bits below (in the register at offset 0x03). The Pointer bits are used to access the Index registers as follows: 0x00 to 0x0F = Qbv Entries Set 1 (defined in register Table 298, Qbv Table Entry, on page 285) 0x10 to 0x1F = Reserved for future use 0x20 = Guard Band 1 Entry (defined in register Table 299, Qbv Table Guard Band, on page 286) 0x21 to 0x3F = Reserved for future use 0x40 to 0x4F = Qbv Entries Set 2 (define in register Table 298, Qbv Table Entry, on page 285) 0x50 to 0x5F = Reserved for future use 0x60 = Guard Band 2 Entry (defined in register Table 299, Qbv Table Guard Band, on page 286) 0x61 to 0x7F = Reserved for future use **NOTE:** The device supports two Qbv Time sets: 1 & 2. The 1st number in the descriptions below are for Set 1 while the 2nd number is for Set 2. Entry 0x00/0x40 of this table is processed on this port at the Global t_0 time (as defined by PTP Trig in PTP TAI Global) plus this port's Port Delay time 1 or 2 (as defined in Qbv Port offset 0x00/0x01). This re-starting of the table entry processing occurs if the entry processing has completed or not. Once an entry is done being processed, the next higher entry number in the table is processed until entry 0x0F/0x4F completes. If entry 0x0F/0x4F completes the default entry is applied. The default entry has a Queue State of 0xFF, Guard Band = 0, and a Window Time = 0x0000. The reset state of this Table causes no queue blocking, effectively disabling Qbv.
7:4	Reserved	RES	Reserved for future use.

Table 297: Qbv Port Table Control (Continued)
Offset: 0x02 or decimal 2 – Qbv Port

Bits	Field	Type	Description
3:0	Queue State	RWS to 0xF	Queue State for Table entry <n> (where <n> = the Pointer bits above). For both the Qbv and Guard Band entries, these bits indicate which egress queues on this port are allowed to participate in the next frame selection for transmission by having these bits presented to the port's egress Scheduler. QPri 0 is controlled by bit 0, QPri 1 is controlled by bit 1, etc., to QPri 3 being controlled by bit 3. When a queue's Queue State bit is set to a one, that queue is eligible to transmit frames. When a queue's Queue State bit is cleared to zero, that queue is blocked (i.e., frames in that queue are not available for transmission).

Table 298: Qbv Table Entry
Offset: 0x03 or decimal 3 – Qbv Port

Bits	Field	Type	Description
15	Guard Band	RWR	Perform Guard Band. When this bit is cleared to a zero, this entry's Queue State bits will be presented to the port's egress Scheduler immediately. Typically used to re-open blocked queues at the end of a window or to perform a simple delay. When this bit is set to a one, the Guard Band's Entry (at Index 0x20/0x60 of this register) is processed first and when that is done, this entry's Queue State bits will be presented to the port's egress Scheduler. Typically used to start a window such that the port is guaranteed idle at the end of the Guard Band (i.e., at the start of this window). **NOTE:** There are two Guard Band registers, one for each Qbv Time set. Guard Band 1 is associated with Qbv Time set 1 and is at index 0x20. Guard Band 2 is associated with Qbv Time set 2 and is at index 0x60.
14:0	Window Time	RWR	Window Time. Once this entry's Queue State bits are presented to the Scheduler, this Window Time is used to define how long this state persists before the next entry in the Qbv Table is processed. The Window Time is in increments of 8ns so it has a maximum time of 262.136 uSec. **NOTE:** A Window Time of all zeros (0x0000) is a special case which indicates the end of the Qbv table has been reached. In this case, this entry's Queue State bits are applied to the port until the next Global t_0 time + Port Delay time at which time processing resumes at entry 0x00 of this table. The Window Time will be interrupted by the next Global t_0 time + Port Delay time (even if it is not all zeros) at which time processing resumes at entry 0x00 of this table. If the last Qbv Entry in the table contains a non-zero Window Time, when this time expires the Queue States will be set to all ones until the next Global t_0 time + Port Delay time event occurs.

These 16-bits are part of each of the Qbv Entries where the other part of each of these entries is the Queue State bits found in the Qbv Table Control and Data register (offset 0x02 above). These fields create a 24-bit Qbv Entry where:

- The Queue State bits define which of the port's Queues are allowed to transmit or not,
- The Window Time bits define how long the above Queue State bits should be applied for, and
- The Guard Band bit defines if the Guard Band time (below) should be applied prior to using the new Queue State bits

Table 299: Qbv Table Guard Band
Offset: 0x03 or decimal 3 – Qbv Port

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:0	GBBytes	RWR	Guard Band Bytes. If a Qbv table entry has its Guard Band bit set to a one, this Guard Band entry is processed prior to that entry being processed. The Guard Band entry is processed by 1st presenting the Guard Band's Queue State bits to the port's egress Scheduler. These Queue State bits will be held for the amount of time it takes to transmit the number of bytes indicated in this entry's GBBytes field. The Guard Band works differently in the following ways: • The period of time it lasts is defined in Tx Byte Time increments. Thus this parameter is port speed dependent! It has a range of 1 to 16383 byte times. • Queue State bits that were set to a one and that are now cleared to a zero by the Guard Band's Queue State bits are considered 'Soft Blocked' • 'Soft Blocked' queues are allowed to start the transmission of a frame as long as it is guaranteed to finish before the end of the Guard Band time is reached. • GBBytes should be set to the size of a maximum size frame bytes to ensure the egress port is idle when the end of the GBBytes time expires. **NOTE:** There are two Guard Band registers, one for each Qbv Time set. Guard Band 1 is associated with Qbv Time set 1 and is at index 0x20. Guard Band 2 is associated with Qbv Time set 2 and is at index 0x60. A GBBytes value of all zeros (0x0000) holds the Guard Band time for ?? Tx byte times (i.e., a few clocks). ?? Tx bytes times is added at the end of each GBByte cycle before the calling Qbv entry is processed. This and the fact that GBByte times is port speed dependent needs to be taken into account when defining the Port's t_0 time. The GBBytes time will be interrupted by the next Port t_0 time at which time processing resumes at entry 0x00 or 0x40 of this table.

These 16-bits are part of Qbv Guard Band Bytes entry where the other part of this entry is the Queue State bits found in the Qbv Table Control and Data register (offset 0x02 above). These fields create a 24-bit Qbv Guard Band Entry where:

- The Queue State bits define which of the port's Queues are 'Soft Blocked' or not,
- The Guard Band Bytes define the size of the 'Soft Blocked' period

Table 300: Qbv Port Debug Register
Offset: 0x08 or decimal 8 – Qbv Port

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:12	GBMode1	RES	Guard Band Mode 1. 0x0 = Queue Sel 1 shows 'hard block' Queue State bits only 0x1 = Queue Sel 1 shows 'soft block' & 'hard block' Queue States 0x2 = Queue Sel 1 shows Guard Band & 'hard block' Queue States 0x3 = Queue Sel 1 shows Guard Band only
11:10	Reserved	RES	Reserved for future use.
9:8	QueueSel1	RWR	Queue Select 1 for Qbv Debug. These bits select which active Qbv Queue State bit on this port is presented to the Qbv Debug Port selection logic for Qbv Debug bit 1. Qbv Debug bit 1 can be presented to any GPIO pin if that GPIO pin is selected to output RX_CLK1/Qbv1 and if the SecRecClkSel (TAI Global offset 0x1E) selects Qbv 1 (its default setting) and this port is selected for Qbv 1 (Qbv Global offset 0x08).
7:6	Reserved	RES	Reserved for future use.
5:4	GBMode0	RES	Guard Band Mode 0. 0x0 = Queue Sel 0 shows 'hard block' Queue State bits only 0x1 = Queue Sel 0 shows 'soft block' & 'hard block' Queue States 0x2 = Queue Sel 0 shows Guard Band & 'hard block' Queue States 0x3 = Queue Sel 0 shows Guard Band only
3:2	Reserved	RES	Reserved for future use.
1:0	QueueSel0	RWR	Queue Select 0 for Qbv Debug. These bits select which active Qbv Queue State bit on this port is presented to the Qbv Debug Port selection logic for Qbv Debug bit 0. Qbv Debug bit 0 can be presented to any GPIO pin if that GPIO pin is selected to output RX_CLK0/Qbv0 and if the PriRecClkSel (TAI Global offset 0x1E) selects Qbv 0 (its default setting) and this port is selected for Qbv 0 (Qbv Global offset 0x08).

The Guard Band Modes allow inclusion of the Guard Band time or the actual soft blocking during the Guard Band time. If the Guard Band is not being use then Guard Band Mode 0 should be used. The Guard Band Mode affects the Queue Sel's as follows:

- 0x0 = Queue Sel will be a 1 during the Guard Band, else it tracks the Queue States
- 0x1 = Queue Sel will be a 1 during the Guard Band if the queue is not 'soft blocked' (i.e., it can transmit a frame), else it tracks the Queue States
- 0x2 = Queue Sel will be a 0 during the Guard Band if the next Queue State is zero, else it tracks the Queue States
- 0x3 = Queue Sel will be a 0 during the Guard Band, else it is a 1

Table 301: Topaz's Qbv Global Register Map
(Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x3 & AVBPort = 0x1F)

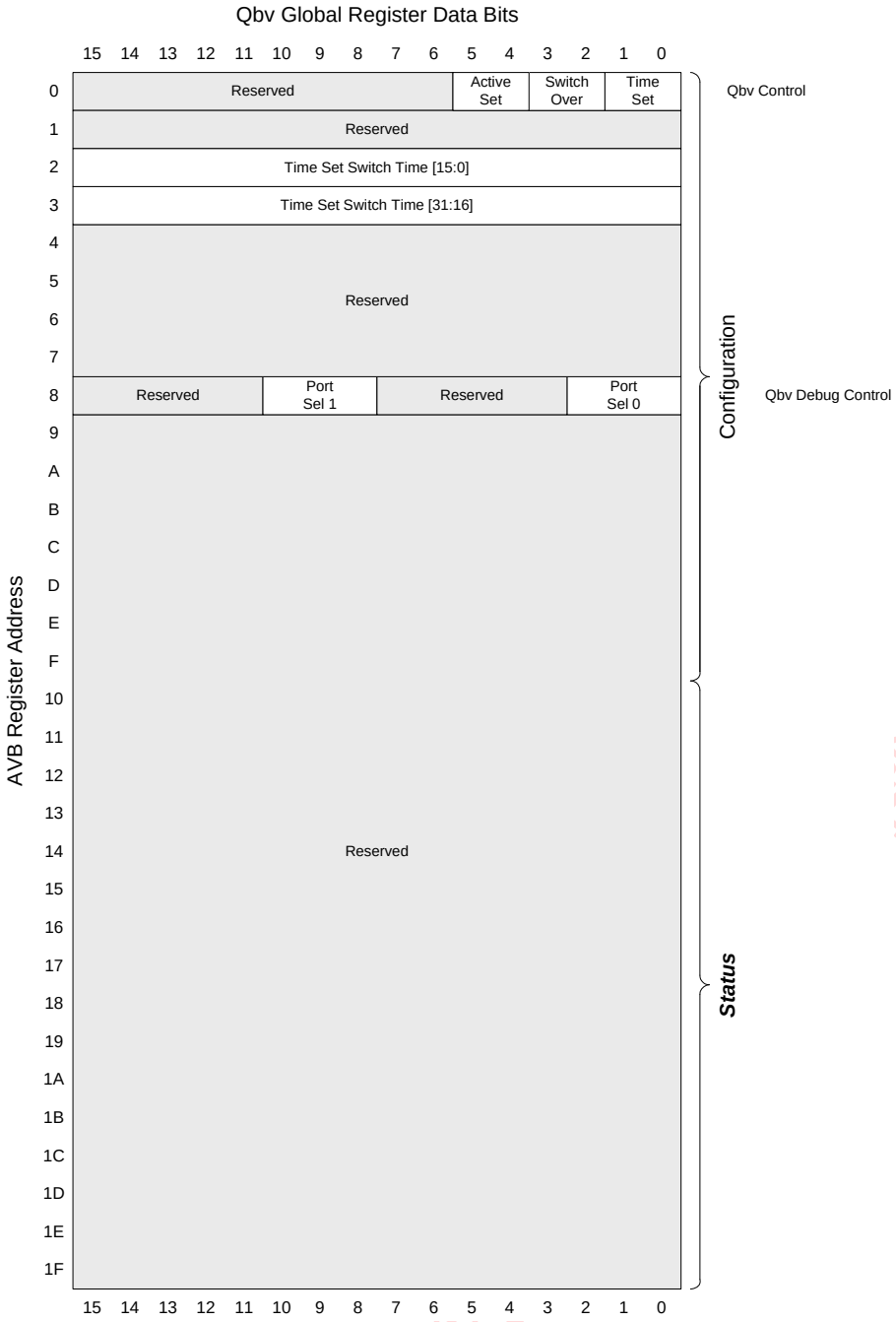


Table 302: Qbv Control Register
Offset: 0x00 or decimal 0 – Qbv Global

Bits	Field	Type	Description
15:2	Reserved	RES	Reserved for future use.
5:4	ActiveTimeSet	RO	Active Qbv Time Set. Two Qbv Time Sets are supported in the device and this register is used to report which Time set is currently active and being used as follows: 0x0 = Default, Time Sets have never toggled or no Set selected 0x1 = Qbv Time Set 1 is active on this port 0x2 = Qbv Time Set 2 is active on this port 0x3 = Reserved for future use. **NOTE:** A value of 0 will appear in this register during the Idle cycle of Switch Over Mode 2 below.

Table 302: Qbv Control Register (Continued)
Offset: 0x00 or decimal 0 – Qbv Global

Bits	Field	Type	Description
3:2	Switch Over Mode	RWR	Qbv Time set Switch Over Mode. Two Qbv Time Sets are supported in the device and this register is used to control how Qbv transitions from one Time Set to another as follows: 0x0 = Transition as soon as possible 0x1 = Transition as soon as possible after Time Set Switch Time occurs 0x2 = Transition after Time Set Switch Time occurs with an idle cycle 0x3 = Mode 2 but where all Queues are blocked during the idle cycle Qbv Switching Mode 0 switches between Time Sets as soon as possible. This mode is used when adding new entries at the end of existing tables. This mode: • Waits for the next Qbv Global window t_0 (as defined by PTP Trig – see TAI registers) • It then uses the new Time Set at this Qbv Global window t_0 Qbv Switching Mode 1 switches between Time Sets up to 32 seconds in the future. This mode allows re-programming an entire network with a coordinated switch over time configured sometime in the future. If more than 32 seconds is needed, software simply waits to program/enable this mode until the last 30 seconds before the desired switch over time. This modes: • Waits for PTP Global Time (TAI offsets 0x0A & 0x0B) to match the Time Set Switch Time registers (Qbv offsets 0x06 & 0x07) • Then it waits for the next Qbv Global window t_0 (as defined by PTP Trig – see TAI registers) • It then uses the new Time Set at this Qbv Global window t_0 Qbv Switching Mode 2 & 3 switches between Time Sets with a gap in-between. This mode is needed if the Time Set's Delay Time plus its accumulative Qbv Table entry times pushes past the next's Qbv Global window t_0. This mode: • Waits for PTP Global Time (TAI offsets 0x0A & 0x0B) to match the Time Set Switch Time registers (Qbv offsets 0x06 & 0x07) • Then it waits for the next Qbv Global window t_0 (as defined by PTP Trig – see TAI registers) • It continues processing the old Time Set if it has not yet completed & if it has completed, it does not start it up again, inserting an idle cycle • Then it waits for the next Qbv Global window t_0 (as defined by PTP Trig – see TAI registers) • It then uses the new Time Set at this Qbv Global window t_0

Table 302: Qbv Control Register (Continued)
Offset: 0x00 or decimal 0 – Qbv Global

Bits	Field	Type	Description
1:0	TimeSet	SC	Qbv Time Set. Two Qbv Time Sets are supported in the device and this register is used to control which time set is active and when as follows: 0x0 = No change from the current Qbv operating mode 0x1 = Switch to Time Set 1 0x2 = Switch to Time Set 2 0x3 = Reserved for future use The transition from Time Sets is handled using the methods enable by the Switching Mode bits above and then these bits return to zero indicating completion of the process.

Table 303: Time Set Switch Time
Offset: 0x02 or decimal 2 – Qbv Global

Bits	Field	Type	Description
15:0	TimeSet SwitchTime [15:0]	RWR	Time Set switch Time Register bits 15 to 0 of a 32-bit register. The TimeSetSwitchTime register is used in multiple ways, but its contents are always relative to the PTP Global Time (aka, H/W Time) found in TAI Global offsets 0x0E & 0x0F. This register is used in Qbv Switch Over Modes 1 & 2 (Qbv offset 0x04). It is used to set a switching time up to 32 seconds in the future. This allows re-programming an entire network with a coordinated switch over time configured sometime in the future. If more than 32 seconds is needed, software simply waits to program/enable this mode until the last 30 seconds before the desired switch over time. The upper 16-bits of this register are contained in the register below.

Table 304: Time Set Switch Time
Offset: 0x03 or decimal 3 – Qbv Global

Bits	Field	Type	Description
15:0	TimeSet SwitchTime [31:16]	RWR	Time Set switch Time Register bits 31 to 16 of a 32-bit register. Value to be matched with global timer value – used to define a Time Set Switch over time sometime in the future. See the description above. The lower 16-bits of this register are contained in the register above.

Table 305: Qbv Global Debug Register
Offset: 0x08 or decimal 8 – Qbv Global

Bits	Field	Type	Description
15:11	Reserved	RES	Reserved for future use.
10:8	PortSel1	RWS to 0x7	Port Select 1 for Qbv Debug. These bits select which Port on this device is presented to the Qbv Debug Port selection logic for Qbv Debug bit 1. Qbv Debug bit 1 can be presented to any GPIO pin if that GPIO pin is selected to output RX_CLK1/Qbv1 and if the SecRecClkSel (TAI Global offset 0x1E) selects Qbv 1 (its default setting). Which Qbv Queue State bit is presented is selected by the Queue Sel 1 bits on the selected port's Qbv registers (Qbv Port offset 0x08). **NOTE:** The default value in this register does not select any port such that the default value of RX_CLK1/Qbv1 is zero.
7:3	Reserved	RES	Reserved for future use.
2:0	PortSel0	RWS to 0x7	Port Select 0 for Qbv Debug. These bits select which Port on this device is presented to the Qbv Debug Port selection logic for Qbv Debug bit 0. Qbv Debug bit 0 can be presented to any GPIO pin if that GPIO pin is selected to output RX_CLK0/Qbv0 and if the PriRecClkSel (TAI Global offset 0x1E) selects Qbv 0 (its default setting). Which Qbv Queue State bit is presented is selected by the Queue Sel 0 bits on the selected port's Qbv registers (Qbv Port offset 0x08). **NOTE:** The default value in this register does not select any port such that the default value of RX_CLK0/Qbv0 is zero.

1.5.6 Switch TCAM Registers – TCAM (88E6341 Only)

The device contains a set of global registers that effect all the Ethernet ports in the device. Each TCAM register is 16-bits wide and their bit assignments are shown in Table 306.

Table 306: Register bit Map for TCAM Page 0 (Device Addr 0x1F)

Global Register Data Bits

SMI Register Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	TB	TCAMOp			0	0	Reserved		TCAM Entry								TCAM Operation ← TB = TCAM Busy
1	Reserved																
2	Mask for Type & SPV[10:8]								Frame Type	Reserved							Page 0 TCAM Keys & Mask
3	Mask for SPV[7:0]									SPV[6:0]							
4	Mask for Res or PPRI/PVID								Reserved or PPRI+PVID[11:8]								
5	Mask for Index or PVID								Index or PVID[7:0]								
6	Mask for bits 7:0 of this register								Frame Octet 1 (DA1) or Octet 49								Page 0 TCAM Match Data & Mask For Bytes 1 to 22 or For Bytes 49 to 70
7	Mask for bits 7:0 of this register								Frame Octet 2 (DA2) or Octet 50								
8	Mask for bits 7:0 of this register								Frame Octet 3 (DA3) or Octet 51								
9	Mask for bits 7:0 of this register								Frame Octet 4 (DA4) or Octet 52								
A	Mask for bits 7:0 of this register								Frame Octet 5 (DA5) or Octet 53								
B	Mask for bits 7:0 of this register								Frame Octet 6 (DA6) or Octet 54								
C	Mask for bits 7:0 of this register								Frame Octet 7 (SA1) or Octet 55								
D	Mask for bits 7:0 of this register								Frame Octet 8 (SA2) or Octet 56								
E	Mask for bits 7:0 of this register								Frame Octet 9 (SA3) or Octet 57								
F	Mask for bits 7:0 of this register								Frame Octet 10 (SA4) or Octet 58								
10	Mask for bits 7:0 of this register								Frame Octet 11 (SA5) or Octet 59								
11	Mask for bits 7:0 of this register								Frame Octet 12 (SA6) or Octet 60								
12	Mask for bits 7:0 of this register								Frame Octet 13 (Tag1) or Octet 61								
13	Mask for bits 7:0 of this register								Frame Octet 14 (Tag2) or Octet 62								
14	Mask for bits 7:0 of this register								Frame Octet 15 (PRI+VID) or Octet 63								
15	Mask for bits 7:0 of this register								Frame Octet 16 (VID) or Octet 64								
16	Mask for bits 7:0 of this register								Frame Octet 17 (Type1) or Octet 65								
17	Mask for bits 7:0 of this register								Frame Octet 18 (Type2) or Octet 66								
18	Mask for bits 7:0 of this register								Frame Octet 19 (Data1) or Octet 67								
19	Mask for bits 7:0 of this register								Frame Octet 20 (Data2) or Octet 68								
1A	Mask for bits 7:0 of this register								Frame Octet 21 (Data3) or Octet 69								
1B	Mask for bits 7:0 of this register								Frame Octet 22 (Data4) or Octet 70								
1C–1F	Reserved																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

1.5.7 Common TCAM Registers for all TCAM pages:

The TCAM registers are used to control the Ternary Content Addressable Memory (TCAM). This large memory is accessed as three separate TCAM pages. Pages 0 & 1 contain the Ingress TCAM's Key and Frame matching data. Page 2 contains the Ingress TCAM Actions that are applied to a frame when the associated Ingress TCAM matching data matches an ingressing frame. Pages 0, 1 & 2 comprise each Ingress TCAM entry as each Ingress TCAM entry has an associated component in each of these Pages. Ingress TCAM entries are Global and can be restricted to be used on one, multiple or all ingress ports.

TCAM offset 0x00 is common to each of the pages while offset 0x01 to 0x1F will be different per page. This section covers the common register at offset 0x00.

Table 307: TCAM Operation Register
Offset: 0x00 or decimal 0 – TCAM Common

Bits	Field	Type	Description
15	TCAMBusy	SC	TCAM Busy. This bit must be set to a one to start a TCAM operation (see TCAMOp below). Only one TCAM operation can be executing at one time so this bit must be zero before setting it to a one. When the requested TCAM operation completes this bit will automatically be cleared to a zero.
14:12	TCAMOp	RWR	TCAM Opcode. The device supports the following TCAM operations (all of these operations can be executed while frames are transiting through the switch): 0x0 = No Operation 0x1 = Flush[1] All entries 0x2 = Flush[2] or invalidate a single TCAM entry 0x3 = Load[3] an entry's page 0x4 = Get Next[4] (read next valid entry) 0x5 = Read[5] an entry's page (performs a direct read of an entry) 0x6 = Reserved 0x7 = Reserved

Table 307: TCAM Operation Register
Offset: 0x00 or decimal 0 – TCAM Common

Bits	Field	Type	Description
11:10	TCAM Page	RWR	TCAM Page. Each Ingress TCAM Entry is made up of 3 pages of data. All three pages need to loaded in a particular order for the Ingress TCAM to operate correctly while frames are flowing through the switch. If the entry is currently valid, it must first be Flushed[6]. Then Page 2 needs to be loaded first, followed by Page 1, and then finally Page 0. Each page load requires its own Write TCAMOp with these TCAM Page bits set accordingly. When Page 0 is loaded the TCAM entry will immediately take affect on the next frame that enters the switch (assuming the TCAM is enabled). When two TCAM entries are being used for 96 byte frame compares[7], the TCAM entry for bytes 49 to 96 needs to loaded first before the TCAM entry for bytes 1 to 48 is entered. The same Page 2, Page 1 then Page 0 process must be used. If these two entries are currently valid, they both must be Flushed prior to the loading of the new data. The TCAM Page bits are also used to address the desired TCAM Page to be loading into TCAM offsets 0x02 through 0x01B for the Read TCAMOp. **NOTE:** These bits must be 0x0 for Flush All TCAMOp. They must be 0x0 for the Get Next and Flush an entry TCAMOps.
9:7	Reserved	RES	Reserved for future use.
6:0	TCAM Entry	RWR	TCAM Entry. These bits must point to the desired Ingress TCAM for Flush an entry (0x2), Load an entry (0x3) and Read an entry (0x5) TCAMOps. These bits return the Ingress TCAM entry found for the Get Next (0x4) TCAMOp.

1. A Flush All command will initialize TCAM Pages 0 and 1, offsets 0x02 to 0x1B to 0x0000, and TCAM Page 2 offsets 0x02 to 0x1B to 0x0000 for all TCAM entries with the exception that TCAM Page 0 offset 0x02 will be initialized to 0x00FF.
2. A Flush a single TCAM entry command will write the same values to a TCAM entry as a Flush All command as long as the command is done to Page 0, but it is done to the selected single Ingress TCAM entry only.
3. The Load sequence of a TCAM entry is critical. See the text for more information.
4. A Get Next operation to Page 0 finds the next higher Ingress TCAM Entry number that is valid (i.e., any entry whose Page 0 offset 0x02 is not equal to 0x00FF). The TCAM Entry register (bits 6:0) is used as the TCAM entry to start from. To find the lowest number TCAM Entry that is valid, start the Get Next operation with TCAM Entry set to 0xFF. When the operation is done the TCAM Entry register contains the next higher valid TCAM entry number and TCAM offsets 0x02 to 0x1B registers contain the data found in that TCAM's entry for the TCAM Page 0. To read back the other pages of this TCAM entry, use the Read TCAMOp to Pages 1 and 2. To find the next higher entry simply issue the Get Next opcode again. If the TCAM Entry register is returned set to all one's and its TCAM Page 0 offset 0x02 is equal to 0x00FF, no higher valid TCAM entry was found (and the end of the table was reached). If the TCAM Entry register is returned set to all one's and its TCAM Page 0 offset 0x02 is not equal to 0x00FF, the highest TCAM entry was found as valid, and the end of the table was reached.
5. A Read TCAMOp loads the TCAM offsets 0x02 to 0x1B registers with the data found in the TCAM entry and its TCAM page pointed to by the TCAM Entry and TCAM Page bits of this register (bits 6:0 and 11:10 respectively).
6. Use the Flush an entry TCAM entry TCAMOp to do this (bits 14:12 of this register).
7. 96 byte compares will result in a TCAM hit only if the ingressing frame is at least 96 bytes in size.

Note: The TCAM Entry value is used to resolve conflicts between multiple Ingress TCAM entry hits for the same ingressing frame. If more than one Ingress TCAM entry matches the contents of a frame, the entry with the lowest TCAM Entry value will be chosen. This requires that the longest or best match TCAM entry needs to have the lowest TCAM Entry value or it won't be chosen when hit.

TCAM entries can be loaded in any order at any location. There is no need to start at 0x00 first and then increment. Flushed or empty TCAM entries can appear anywhere in the table, as can valid TCAM entries.

1.5.8 TCAM Registers for TCAM Page 0:

Ingress TCAM Key Data and Frame Match Data bytes are in the lower 8 bits of each 16-bit register. The upper 8 bits of each register are the Mask bits for the lower 8 bits where bit 15 is the mask for bit 7, bit 14 is the mask for bit 6, etc. The individual pairs of data bits and mask bits work together as follows:

Mask	Data	Meaning
0	0	Don't Care. The data bit can be a one or a zero for a TCAM hit to occur.
1	0	Hit on 0. The data bit must be a zero for a TCAM hit to occur.
1	1	Hit on 1. The data bit must be a one for a TCAM hit to occur.
0	1	Never Hit. Used to prevent a TCAM hit from occurring from this entry[1].

The Never Hit value is used to Flush the TCAM or Purge a TCAM entry. On a TCAM Flush or Purge, this value it written to the 1st TCAM byte only (offset 0x02 on TCAM page 1). On a TCAM Flush All or on a TCAM Flush an entry, all other TCAM data and mask bytes are written to a value of 0x0000 and so are the Action bytes.

1. This is needed so that a TCAM entry can be defined as unused or invalid.

Table 308: Keys Register 1
Offset: 0x02 or decimal 2 – TCAM Page 0

Bits	Field	Type	Description
15:14	Mask	RWR	Mask for bits 7:6 below. Both of these bits must be zeros for unused entries and must be ones for valid entries.
13:8	Reserved	RWR	Reserved for future use. These bits must always be zeros.
7:6	Frame Type	RWS	Frame Type. These bits are used to define the Frame type or mode this TCAM entry is defined for as follows: For bytes 1 to 48 of a TCAM Entry: 0x0 = Normal Network frame, 0x8100 tagged or untagged 0x1 = DSA Tagged[1] (this selection changes the definition of the tag bytes – TCAM Page 1 offsets 0x12 to 0x15) 0x2 = Provider Tagged[2] (this selection changes the definition of TCAM Page 1 offsets 0x04 and 0x05) 0x3 = Reserved for future use (except to indicate this TCAM entry is unused if bits 15:14 are zero). Do not use this setting if bits 15:14 are ones. For bytes 49 to 96 of a TCAM Entry: 0x0 = Continued Entry[3] for bytes 49 to 96 0x1 = Reserved 0x2 = Reserved 0x3 = Reserved **NOTE:** DSA Tagged mode should only be used on TCAM entries that are associated to ports that are in DSA or Ether Type DSA mode (see FrameMode in Port offset 0x04). Likewise Provider Tagged mode should only be used on TCAM entries that are associated to ports that are in Provider mode. It can be useful for Normal Network mode to be associated to a Provider port – that way non-Provider tagged frames that enter that Provider port can be processed by that TCAM entry. Normal Network mode entries are also useful on Ether Type DSA tag mode ports. **NOTE:** These TCAM Entry Frame Type bits are matched to the resulting mode each frame is determined to be as it is being received. This is directly related to the Port's Frame Mode setting (Port offset 0x04) as follows: Port's Frame Mode — Possible Frame Types on 1st 48 Byte TCAM compares 0x0 = Normal — 0x0 = Normal 0x1 = DSA — 0x1 = DSA 0x2 = Provide — 0x2 = Provider (if Provider Tagged) else 0x0 = Normal 0x3 = DSA — 0x1 = DSA (if Ether Type DSA) else 0x0 = Normal

Table 308: Keys Register 1
Offset: 0x02 or decimal 2 – TCAM Page 0

Bits	Field	Type	Description
5:0	Reserved but special – read description	RWS	Reserved for future use. These bits must be ones for unused entries and must be zeros for valid entries.

1. Ethertype DSA tagged frames get their Ether Type and following 2 null bytes removed during ingress. Therefore the rest of the TCAM match data must match the standard DSA frame format.
2. Provider Tagged frames get their Provider Tag (or multiple Provider Tags if there are more than one) removed during ingress. Therefore the rest of the TCAM match data must match the modified frame without the Provider Tag(s). The 1st Provider Tag that was removed can have its tag data matched in the TCAM – see TCAM Page 1 offsets 0x04 and 0x05.
3. A Continued Entry has a non-zero Index value – see Page 0 offset 0x05.

Table 309: Keys Register 2
Offset: 0x03 or decimal 3 – TCAM Page 0

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:8	Mask	RWR	Mask for bits 7:0 of this register. All valid TCAM entries must have its SPV mask bits set to a one for the ports that are specifically not to be associated with this entry, see the SPV bit description below.
7	Reserved	RES	Reserved for future use.
6:0	SPV	RWR	Source Port Vector. These bits are used to define which switch ports can use this TCAM entry. This way one TCAM entry can be associated with more than one port if those ports are to take the same Action for the same kind of frame. Bit 0 is associated with Port 0, bit 1 with Port 1, etc. Use the SPV and Mask bits as follows: If a TCAM entry is to be associated with only one port then set all the SPV Mask bits to a one and set the SPV to the bit vector of the port. For example: If an entry is to be associated with Port 5 only, then set the SPV Mask bits above to 0x7F and the SPV bits in this register to 0x20. Alternatively the method below can be used for a single port as well. If a TCAM entry is to be associated with more than one port but specifically excluded from other ports, then set all the excluded port's SPV Mask bits above to one and set the SPV bits in this register to all zeros. For example: If an entry is to be associated with Port 2 and Port 5 only, then set the SPV Mask bits to 0x5B and the SPV bits to 0x00. What is really being done here is specifically excluding specific ports from using this entry as their bits must be zero for this entry to match. If a TCAM entry is to be associated with all the ports in the device then set the SPV bits to a don't care value by setting the SPV Mask bits and the SPV bits in this register to 0x00. **NOTE:** A port must be enabled to use the TCAM (Port offset 0x0D) or that port's bits in this register will have no effect. These bits are matched to a vectorized conversion of the ingressing frame's physical source port number.

Table 310: Keys Register 3
Offset: 0x04 or decimal 4 – TCAM Page 0

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:4	Reserved or PPRI	RWR	Reserved for Provider PRI. Reserved for future use when the TCAM entry's FrameMode bits (TCAM Page 0, offset 0x02) are not equal to Provider Tagged or if this is a TCAM entry for searches beyond the first 48 bytes. When the TCAM entry's FrameMode bits are Provider Tagged, these bits are Provider Priority bits. When a port is in Provider Mode (see FrameMode, Port offset 0x04), and if the ingressing frame is provider tagged, the provider tag(s) are removed from the frame and the 1st provider tag's data is saved. That saved data is compared to these bits. Bits 7:5 are the frame's Provider PRI bits and bit 4 is the frame's Provider DE (Drop Eligible) bit.
3:0	Reserved or PVID[11:8]	RWR	Reserve for Provider VID[11:8]. Reserved for future use when the TCAM entry's FrameMode bits (TCAM Page 0, offset 0x02) are not equal to Provider Tagged or if this is a TCAM entry for searches beyond the first 48 bytes. When the TCAM entry's FrameMode bits are Provider Tagged, these bits are Provider VID[11:8] bits. When a port is in Provider Mode (see FrameMode, Port offset 0x04), and if the ingressing frame is provider tagged, the provider tag(s) are removed from the frame and the 1st provider tag's data is saved. That saved data is compared to these bits. Bits 3:0 are the frame's Provider VID[11:8] bits.

Table 311: Keys Register 4
Offset: 0x05 or decimal 5 – TCAM Page 0

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Index or PVID[7:0]	RWR	Index or Provider VID[7:0]. When the TCAM entry's FrameMode bits are Provider Tagged, these bits are Provider VID[7:0] bits for TCAM entries for the first 48 bytes of a frame only. When a port is in Provider Mode (see FrameMode, Port offset 0x04), and if the ingressing frame is provider tagged, the provider tag(s) are removed from the frame and the 1st provider tag's data is saved. That saved data is compared to these bits. Bits 7:0 are the frame's Provider VID[7:0] bits. When the TCAM entry's FrameMode bits are not equal to Provider Tagged or if this is a TCAM entry for TCAM searches beyond the first 48 bytes, these bits are the Index bits. Index is used to concatenate two TCAM entries to form a 96 byte frame data lookup. If a TCAM entry only needs to look at the first 48 bytes of a frame, then these bits must be 0x00 (with a Mask of 0xFF). If a TCAM entry needs to support a 96 byte lookup, two TCAM entries are required. Load the TCAM match data for bytes 49-96 first with a unique[1], non-zero number in this Index field. A recommended number to use is that entry's TCAM Entry number (TCAM offset 0x00) as that will be a unique number. Then load the TCAM match data for bytes 1 to 48 with this field cleared to 0x00, set that entry's Continue Action bit to a one (TCAM page 2 offset 0x02), and place the value of the upper match entry's Index field into the lower match entry's Next Index field (TCAM page 2 offset 0x03).

1. The TCAM entry that is matching the frame's upper data (bytes 49-96) must have a non-zero Index value. That ensures it will not become a hit when the 1st 48 bytes of the frame are being matched (the 1st 48 byte matches of a frame always assert a 0x00 value to compare with this Index field). If the 1st 48 byte compare returns a 'hit' that needs to be extended to frame bytes 49 to 96, it signals this by setting its Continue Action register bit to a one (TCAM Page 2 offset 0x02). The connection between the 1st 48 byte match data and the 2nd 48 bytes is done through the Next Index data. The 1st 48 byte compare is the 1st part of this connection. It must return a non-zero value in its Next Index data (TCAM Page 2 offset 0x03). That Next Index data is then used as the data to match to the Index field for the upper 48 byte TCAM entry or entries when comparing frame data bytes 49 to 96. If a 'hit' occurs in that 2nd TCAM lookup, the actions from that 2nd 'hit' are used for the frame. If there is no 'hit' for bytes 49 to 96, then there is no 'hit' for the frame (even though there might have been a 'hit' for the 1st 48 bytes of the frame). The Index value for 96 bytes compares don't have to be unique if there more than one upper TCAM entries that would be valid as possible 'hits' to the same lower 48 byte match. As this is hard to think about, it is recommended that a unique value be used for the Index and that its value be set to the value of the upper entries TCAM Entry value (i.e., where it is actually stored in the TCAM).

Table 312: Match Data Register 1
Offset: 0x06 or decimal 6 – TCAM Page 0

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 1 or 49	RWR	Frame Octet 1 or 49. This is the match data for octet 1 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 49.

Table 313: Match Data Register 2
Offset: 0x07 or decimal 7 – TCAM Page 0

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 2 or 50	RWR	Frame Octet 2 or 50. This is the match data for octet 2 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 50.

Match Data Register 3 through 22 continue on in the same way supporting match data for frame byte 3 (or 51) up to frame byte 22 (or 70). These registers are found in TCAM Page 0 offsets 0x08 to 0x1B. All match bytes work the same way.

The Tag match bytes (frame bytes 13 to 16 or TCAM Page 0 offsets 0x12 to 0x15) need to be configured as follows based on the desired affect:

- For TCAM entries that can 'hit' on Normal Network frames that are either 802.1Q Tagged or Untagged: Set the entry's FrameMode (TCAM Page 0, offset 0x02) to 0x0 and set all four Tag match bytes (TCAM Page 0, offsets 0x12 to 0x15) to 0x0000 (don't care). The TCAM compare logic takes Untagged frames and shifts them up by four bytes starting after the frame's Source Address field so the same TCAM entry can match both Tagged and Untagged frames.
- For TCAM entries that can 'hit' on Normal Network frames that are Untagged only: Set the entry's FrameMode (TCAM Page 0, offset 0x02) to 0x0 and set all four Tag match bytes (TCAM Page 0, offsets 0x12 to 0x15) to 0xFF00 (match zeros). The TCAM compare logic takes Untagged frames and shifts them up by four bytes starting after the frame's Source Address field and inserts four bytes of zero. Tagged frames won't look this way.
- For TCAM entries that can 'hit' on Normal Network frames that are Tagged only: Set the entry's FrameMode (TCAM Page 0, offset 0x02) to 0x0 and set the upper two Tag match bytes (TCAM Page 0, offsets 0x12 to 0x13) to 0xFF81 and 0xFF00 respectively (match 0x8100). Only Tagged frames look this way. Set the lower two Tag match bytes (TCAM Page 0, offset 0x14 to 0x15) to don't care (0x00FF) if the value of the Tag is important. Otherwise the TCAM entry can be set to match part or all of Tag's data bits.
- For TCAM entries that can 'hit' on Provider Tagged only: Set the entry's FrameMode (TCAM Page 0, offset 0x02) to 0x2. The Tag match bytes (TCAM Page 0, offset 0x12 to 0x15) are not used to match to the Provider Tag data as they are used to match a Customer tag that might exist in the frame (as defined above). If the TCAM entry needs to match to parts of the Provider Tag data, this data can be matched using Key register 3 and 4 (TCAM Page 0, offset 0x04 and 0x05).
- For TCAM entries that can 'hit' on DSA Tagged only: Set the entry's FrameMode (TCAM Page 0, offset 0x02) to 0x1. The four Tag match bytes (TCAM Page 0, offset 0x12 to 0x15) are used to match to the DSA Tag data where enabled.

1.5.9 TCAM Registers for TCAM Page 1 – Frame Match Data:

Table 314: Register bit Map for TCAM Page 1 (Device Addr 0x1F)

Global Register Data Bits

SMI Register Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	TB	TCAMOp			0	1	Reserved			TCAM Entry							0
1	Reserved																1
2	Mask for bits 7:0 of this register						Frame Octet 23 (Data7) or Octet 71										2
3	Mask for bits 7:0 of this register						Frame Octet 24 (Data7) or Octet 72										3
4	Mask for bits 7:0 of this register						Frame Octet 25 (Data7) or Octet 73										4
5	Mask for bits 7:0 of this register						Frame Octet 26 (Data8) or Octet 74										5
6	Mask for bits 7:0 of this register						Frame Octet 27 (Data9) or Octet 75										6
7	Mask for bits 7:0 of this register						Frame Octet 28 (Data10) or Octet 76										7
8	Mask for bits 7:0 of this register						Frame Octet 29 (Data11) or Octet 77										8
9	Mask for bits 7:0 of this register						Frame Octet 30 (Data12) or Octet 78										9
A	Mask for bits 7:0 of this register						Frame Octet 31 (Data13) or Octet 79										10
B	Mask for bits 7:0 of this register						Frame Octet 32 (Data14) or Octet 80										11
C	Mask for bits 7:0 of this register						Frame Octet 33 (Data15) or Octet 81										12
D	Mask for bits 7:0 of this register						Frame Octet 34 (Data16) or Octet 82										13
E	Mask for bits 7:0 of this register						Frame Octet 35 (Data17) or Octet 83										14
F	Mask for bits 7:0 of this register						Frame Octet 36 (Data18) or Octet 84										15
10	Mask for bits 7:0 of this register						Frame Octet 37 (Data19) or Octet 85										16
11	Mask for bits 7:0 of this register						Frame Octet 38 (Data20) or Octet 86										17
12	Mask for bits 7:0 of this register						Frame Octet 39 (Data21) or Octet 87										18
13	Mask for bits 7:0 of this register						Frame Octet 40 (Data22) or Octet 88										19
14	Mask for bits 7:0 of this register						Frame Octet 41 (Data23) or Octet 89										20
15	Mask for bits 7:0 of this register						Frame Octet 42 (Data24) or Octet 90										21
16	Mask for bits 7:0 of this register						Frame Octet 43 (Data25) or Octet 91										22
17	Mask for bits 7:0 of this register						Frame Octet 44 (Data26) or Octet 92										23
18	Mask for bits 7:0 of this register						Frame Octet 45 (Data27) or Octet 93										24
19	Mask for bits 7:0 of this register						Frame Octet 46 (Data28) or Octet 94										25
1A	Mask for bits 7:0 of this register						Frame Octet 47 (Data29) or Octet 95										26
1B	Mask for bits 7:0 of this register						Frame Octet 48 (Data30) or Octet 96										27
1C	Reserved																28
1D																	29
1E																	30
1F																	31
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

TCAM Operation ← TB = TCAM Busy

P age 1
TCAM Match Data & Mask
For Bytes 23 to 48 or
For Bytes 71 to 96

Table 315: TCAM Operation Register
Offset: 0x00 or decimal 0 – TCAM Common

Bits	Field	Type	Description
15	TCAMBusy	SC	TCAM Busy. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
14:12	TCAMOp	RWR	TCAM Opcode. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
11:10	TCAM Page	RWR	TCAM Page. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
9:7	Reserved	RES	Reserved for future use.
6:0	TCAM Entry	RWR	TCAM Entry. This register is the same across TCAM pages 0, 1 and 2. Please see this register's full description under TCAM Page 0.

Table 316: Match Data Register 23
Offset: 0x02 or decimal 2 – TCAM Page 1

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 23 or 71	RWR	Frame Octet 23 or 71. This is the match data for octet 23 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 71.

Table 317: Match Data Register 24
Offset: 0x03 or decimal 3 – TCAM Page 1

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 24 or 72	RWR	Frame Octet 24 or 72. This is the match data for octet 24 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 72.

Match Data Register 25 through 48 continue on in the same way supporting match data for frame byte 24 (or 73) up to frame byte 48 (or 96. These registers are found in TCAM Page 1 offsets 0x04 to 0x1B. All match bytes work the same way.

1.5.10 TCAM Registers for TCAM Page 2 – Ingress Actions:

Table 318: Register bit Map for TCAM Page 2 (Device Addr 0x1F)

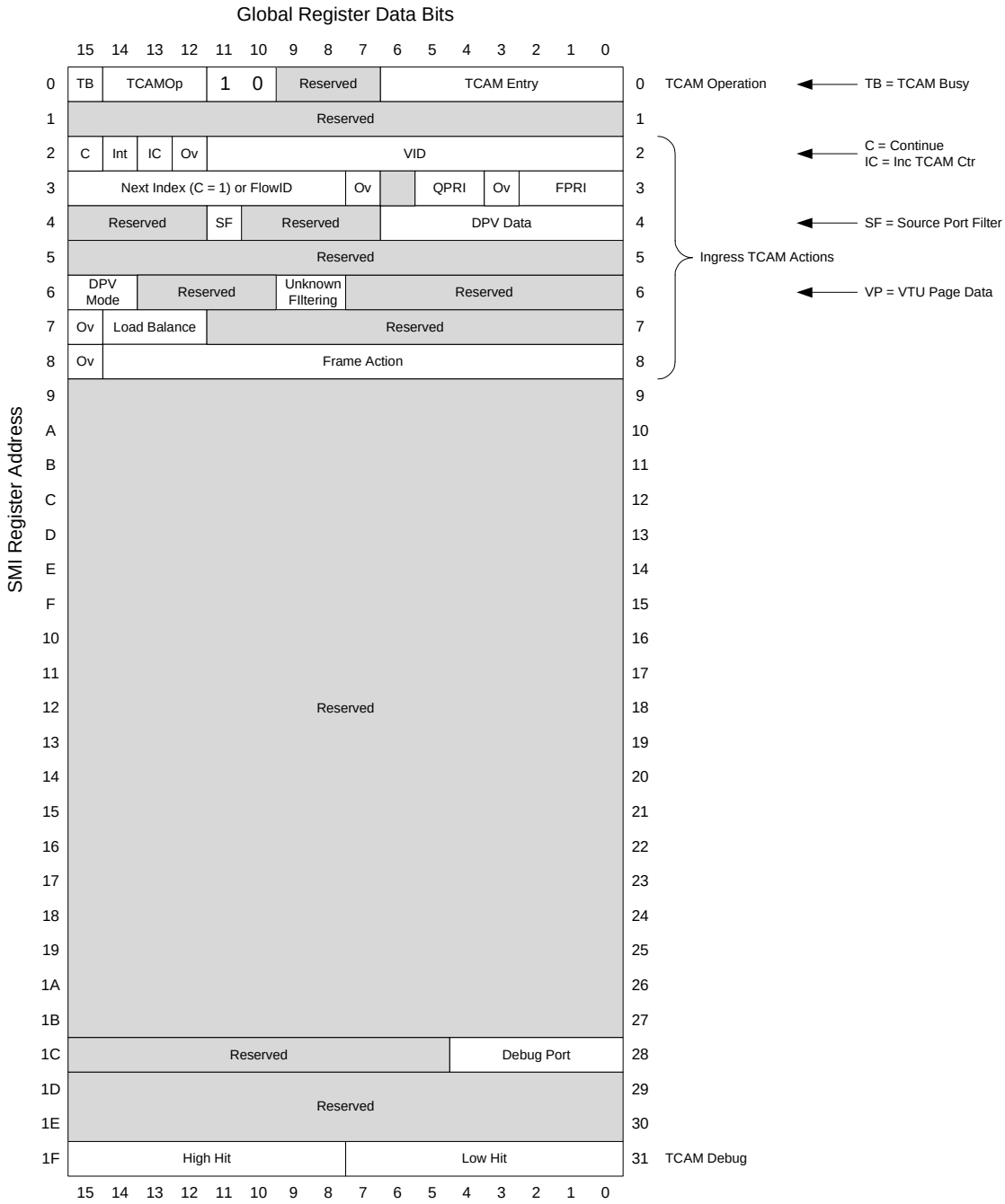


Table 319: TCAM Operation Register
Offset: 0x00 or decimal 0 – TCAM Common

Bits	Field	Type	Description
15	TCAMBusy	SC	TCAM Busy. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
14:12	TCAMOp	RWR	TCAM Opcode. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
11:10	TCAM Page	RWR	TCAM Page. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
9:7	Reserved	RES	Reserved for future use.
6:0	TCAM Entry	RWR	TCAM Entry. This register is the same across TCAM pages 0, 1 and 2. Please see this register's full description under TCAM Page 0.

Table 320: Action Register 1
Offset: 0x02 or decimal 2 – TCAM Page 2

Bits	Field	Type	Description
15	Continue	RWR	Continue this TCAM entry. If this TCAM entry only needs to support a compare in the first 48 bytes of a frame, or if this is the TCAM entry covers bytes 49 to 96 of a frame, set this bit to a zero. This bit should only be a 1 on TCAM entries that cover the first 48 bytes of a frame that needs to be extended to also match bytes 49 to 96 of the frame. See the Next Index bits in Action Register 2, offset 0x03.
14	Int	RWR	Interrupt on a TCAM hit. When this bit is set to a one on a TCAM entry (where the Continue bit is a zero), a TCAM hit interrupt will be generated whenever a match occurs to this entry. The results of the interrupt are stored in the TCAM Int bit (Global 1, offset 0x00).
13	IncTcamCtr	RWR	Increment the port's TCAM Counter pointed to by FlowID[7:6] on a TCAM hit. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the port where the frame ingressed that generated this TCAM hit will get one of the port's TcamCtr[3:0] counters (Global 1 offset 0x1D) incremented by 1. The counter that gets incremented is determined by the TCAM entry's FlowID[7:6] bits (TCAM Page 2, offset 0x03). If FlowID[7:6] = 0 the port's TcamCtr[0] will be incremented, if FlowID[7:6] = 1 the port's TcamCtr[1] will be incremented, etc.
12	VID Override	RWR	VID Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the VID Data (bits 11:0 below) are assigned to the frame overriding any other VID assignment in the switch.
11:0	VID Data	RWR	VID Override Data. When the VID Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this VID Data is assigned to the frame. This VID data is used for ATU Lookups and Learns and it is the VID that will appear in the frame if the frame egresses tagged.

Table 321: Action Register 2
Offset: 0x03 or decimal 3 – TCAM Page 2

Bits	Field	Type	Description
15:8	Next Index or Flow ID	RWR	Next Index or Flow ID. When the Continue bit (TCAM Page 2, offset 0x02) is a one, it means that this TCAM 'hit' is not done yet. A subsequent TCAM search needs to be done with the next 48 bytes of the frame (up to byte 96). To distinguish the subsequent search from the first search, the subsequent search is done with this Next Index data being use as the next search's Index key (TCAM Page 0, offset 0x05). When the Continue bit (TCAM Page 2, offset 0x02) is a zero, these bits are the Flow ID. FlowID is used as follows: Flow ID[1:0] is decoded and sent to the Port Ingress Rate Limiter (PIRL – Global 2 offsets 0x09 & 0x0A) such that up to 4 flows per port can be limited and/or counted separately as long a FlowID[5] = 0 (see below). FlowID[4:0] is placed into the Egress Header when Header Type = 0x0 (Global 1 offset 0x1C). This way the receiving device, typically a CPU, can use these bits as a Flow indicator. FlowID[4:0] will be zeros in the Egress Header on all TCAM misses. FlowID[5] = 0 validates FlowID[1:0] to PIRL. If FlowID[5] = 1 then FlowID[1:0] are ignored by PIRL. FlowID[7:6] is used to indicate which of the ingress port's four TcamCtr's to increment when the IncTcamCtr bit = 1 (TCAM Page 2, offset 0x02).
7	QPRI Override	RWR	QPRI Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the QPRI Data (bits 5:4 below) are assigned to the frame overriding any other QPRI assignment in the switch.
6	Reserved	RES	Reserved for future use.
5:4	QPRI Data	RWR	QPRI Override Data. When the QPRI Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this QPRI Data is assigned to the frame. This QPRI data is used to determine which internal switch queue the frame should be mapped into.
3	FPRI Override	RWR	FPRI Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the FPRI Data (bits 2:0 below) are assigned to the frame overriding any other FPRI assignment in the switch.
2:0	FPRI Data	RWR	FPRI Override Data. When the FPRI Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this FPRI Data is assigned to the frame. This FPRI data is the PRI that will appear in the frame if the frame egresses tagged.

Table 322: Action Register 3
Offset: 0x04 or decimal 4 – TCAM Page 2

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11	Source Port Filter	RWR	Source Port Filter. When this bit is set to a one frames will not egress back out the source port of the frame even if the DPV is modified by the TCAM Entry's DPV Data bits (below). Since each TCAM entry can be used on more than one source port the modification of the DPV by the TCAM could result in frames being mapped back out the source port. This bit being set to a one prevents that as TCAM Source Port Filtering is the last step performed and it is after any DPV modifications done by the TCAM's DPV Data bits.
10:7	Reserved	RES	Reserved for future use.
6:0	DPV Data	RWR	DPV Data. When the DPV Mode bits (offset 0x06 below) are non-zero on a TCAM entry that is hit (where the Continue bit is a zero), this DPV Data is used to modify the DPV assigned to the frame. This DPV Data is used to modify the mapping of the frame, i.e., which port or ports this frame egresses. Bit 0 is used to modify the frame's mapping to Port 0. Bit 1 is used to modify the frame's mapping to Port 1, etc. Based on the non-zero value of the DPV Mode bits the frame's DPV can be overridden, logically OR'ed or logically AND'ed. See the description of the DPV Mode bits below (offset 0x06). **NOTE:** Source Port filtering will be done if the Source Port Filter bit (above) is set to a one.

Table 323: Action Register 5
Offset: 0x06 or decimal 6 – TCAM Page 2

Bits	Field	Type	Description
15:14	DPV Mode	RWR	DPV Mode. These bits control how the DPV Data bits above (offset***s*** 0x04 ***& 0x05***) are applied to the DPV computed by the switch's ingress process as follows: 0x0 = Do Nothing – the frame's DPV is not modified 0x1 = Bit wise AND the frame's DPV with the TCAM Entry's DPV Data 0x2 = Bit wise OR the frame's DPV with the TCAM Entry's DPV Data 0x3 = Replace the frame's DPV with the TCAM Entry's DPV Data Option 0 (Do Nothing) is used to allow the switch mapping to be used on the frame as is (but allow the TCAM to modify other aspects of the frame). Option 1 (AND the DPV) is use to prevent this frame from egressing certain ports (those ports that are zero in the TCAM Entry's DPV). Option 2 (OR the DPV) is used to map this frame to a port (or ports) that it otherwise may not already be mapped to (those ports that are one in the TCAM Entry's DPV). Option 3 (Replacing the DPV) is used to re-map where a frame goes including causing the frame to be discarded by setting the new DPV to all zeros. **NOTE:** If the resulting DPV that is assigned to the frame is all zeros, the frame will be Filtered and the Port's InFiltered counter will increment accordingly. In this case, to prevent this filtered frame from being mirrored, this TCAM entry must also set the FAction Override to a one, with the FAction Data bits all zeros (offset 0x07). **NOTE:** If the resulting DPV maps a frame back to the frame's source port the frame *will* egress its source port unless the Source Port Filter bit (offset 0x04) is set to a one.
13:10	Reserved	RES	Reserved for future use.
9:8	Unknown Filtering	RWR	Unknown Filtering. Bit 9 of this register is used for Multicast Unknown Filtering. Bit 8 is used for Unicast Unknown Filtering. Multicast Unknown Filtering includes all frames whose Mulicast bit set to a one in the frame's DA, but specifically does not include the special Broadcast DA. When an Unknown Filtering bit is set to a one and the frame's DA matches the rules defined above, and that frame's DA is not found present in the address data base (ATU) then that frame will be discarded (filtered). **NOTE:** A special bit for Broadcast Unknowns is not supported in the TCAM as a simple TCAM match for the specific fields is quite easy.

Table 323: Action Register 5
Offset: 0x06 or decimal 6 – TCAM Page 2

Bits	Field	Type	Description
7:0	Reserved	RES	Reserved for future use.

Table 324: Action Register 6
Offset: 0x07 or decimal 7 – TCAM Page 2

Bits	Field	Type	Description
15	Load Balance Override	RWR	Load Balance Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the Load Balance Data (bits 2:0 below) are assigned to the frame overriding any other Load Balance assignment in the switch.
14:12	Load Balance Data	RWR	Load Balance Override Data. When the Load Balance Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this Load Balance Override Data is assigned to the frame. This Load Balance data is used to map which port or ports a frame egresses. This data is used to access the LAG Masking Table (Global 2, offset 0x07) for this frame instead of using the results from the frame's DA & SA. **NOTE:** Source Port filtering will be done.
11:0	Reserved	RES	Reserved for future use.

Table 325: Action Register 7
Offset: 0x08 or decimal 8 – TCAM Page 2

Bits	Field	Type	Description
15	FAction Override	RWR	Frame Action Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the Frame Action Data (bits 14:4 below) are assigned to the frame overriding any other Frame Action assignment in the switch.
14:0	FAction Data	RWR	Frame Action Override Data. When the Frame Action Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this Frame Action Data is assigned to the frame. The Frame Action bits are assigned as follows: 14 – Source is Tagged 13 – ProviderVID 12 – MGMT 11 – ARP 10 – Snoop 9 – PolMirror 8 – PolTrap 7 – SaAvbNRL 6 – DaAvbNRL 5 – DontLearn 4 – Reserved 3 – Reserved 2 – Reserved 1 – GoodAVB (Only valid for devices that support AVB/TSN) 0 – Reserved The above Frame Action data is assigned to the frame at the end of the Ingress Processing. This updated data is then passed to the Port Ingress Rate Limiter (IRL) and the Queue Controller (QC). IRL can be programmed to look at MGMT, ARP, PolMirror, PolTrap, SaNRL and DaNRL bits and in the process limit how many of these frames are allowed to ingress a port or let some frames bypass their normal limiting. The QC also uses these bits to modify the frame's processing as follows: **Source is Tagged (Src_Tag)** is used to indicate if an ingressing frame is Tagged (Src_Tag = 1) or not. The egress logic uses this information to know how to modify the frame when it is directed to transmit the frame Tagged or UnTagged. DSA egress ports also use this information to add the DSA tag or to convert the frame's existing tag to a DSA tag. Forcing Src_Tag to zero will cause a tag (DSA or normal) to be inserted even if the ingressing frame was tagged. Don't force Src_Tag to a one on untagged frames. **ProviderVID *(PTag)*** causes the inserted Provider Tag (on Provider Tagged egress ports) to get the frame's VID from the assignment made in Ingress instead of using the source Port's DefaultVID (Port offset 0x07). This bit must be set to a 1 if this TCAM entry is assigning a new VID for frames that egress out a Provider Port – or that new VID will not be in the Provider's Tag.

Table 325: Action Register 7 (Continued)
Offset: 0x08 or decimal 8 – TCAM Page 2

Bits	Field	Type	Description
14:0 (cont.)	FAction Data	RWR	**MGMT** allows frames to egress Blocked ports (Port offset 0x04) and prevents the use of tagging information from the VTU (i.e., the frame will egress unmodified unless the port is a Header or DSA port). It sets the MGMT bit in the egress Marvell Header. If the frame egresses a DSA port the frame will be marked as a To_CPU frame with a MGMT CPU Code. **ARP** only has a effect if ARP mirroring is enabled on the ingress port (Port offset 0x08). If that is the case, it will cause this frame to be copied to the port indicated by CPU Dest (Global 1, offset 0x1A). If the CPU Dest port's FrameMode is a DSA mode (Port offset 0x04) the copied frame will egress with a To_CPU tag with an APR CPU Code. ARP will become a Trap (instead of a Mirror) if the DPV of the frame is zeros. **Snoop** allows frames to egress Blocked ports (Port offset 0x04) and prevents the use of tagging information from the VTU (i.e., the frame will egress unmodified unless the port is a Header or DSA port). It sets the Snoop bit in the egress Marvell Header. If the frame egresses a DSA port the frame will be marked as a To_CPU frame with an IGMP CPU Code. **PolMirror** it will cause this frame to be copied to the port indicated by Mirror Dest (Global 1, offset 0x1A). If the Mirror Dest port's FrameMode is a DSA mode (Port offset 0x04) the copied frame will egress with a To_CPU tag with a PolMirror CPU Code. PolMirror will become a Trap (instead of a Mirror) if the DPV of the frame is zeros. **PolTrap** allows frames to egress Blocked ports (Port offset 0x04) and prevents the use of tagging information from the VTU (i.e., the frame will egress unmodified unless the port is a Header or DSA port). It sets the PolTap bit in the egress Marvell Header. If the frame egresses a DSA port the frame will be marked as a To_CPU frame with a PolTrap CPU Code. **SaAvbNrl** and **DaAvbNrl** do not modify the frame's content in any way but do override the normal source of these signals to PIRL. PIRL can then be programmed to use these signals as an indication of what frames can bypass PIRL's normal frame rate limiting (see PRIL Bucket Configuration, offset 0x07). **DontLearn** prevents learning from occurring on this frame. **Good_AVB** prevents the frame's FPRI from being mapped away from AVB FPRI's on AVB enabled egress ports. **NOTE:** All of these Frame Actions are sent to IRL where they can be used to limit this frame or used to prevent its limiting.

Table 326: TCAM Debug Register
Offset: 0x1C or decimal 28 – TCAM Page 2

Bits	Field	Type	Description
15:4	Reserved	RES	Reserved for future use.
3:0	Debug Port	RWR	Debug Port number. This register is used to define the port number whose TCAM activities will be tracked in the TCAM Debug registers in offset 0x1F below.

Table 327: TCAM Debug Register
Offset: 0x1F or decimal 31 – TCAM Page 2

Bits	Field	Type	Description
15:8	High Hit	RO	TCAM Entry for High 48-byte Hit. This register gets updated with the TCAM Entry number that was a 'hit' for any compare to frame bytes 49 to 96 for the port number defined in offset 0x1C above. This register will get overwritten with every new 'hit' for frame bytes 49 to 96 and it gets written to zeros on every Low Hit update.
7:0	Low Hit	RO	TCAM Entry for Low 48-byte Hit. This register gets updated with the TCAM Entry number that was a 'hit' for any compare to frame bytes 1 to 48 for the port number defined in offset 0x1C above. This register will get overwritten with every new 'hit' for frame bytes 1 to 48. It gets written to ones at reset or if there is not a match (a 'miss') for any compare to frame bytes 1 to 48.

The above registers work as follows in the following example sequence:

- 0x00FF = Reset state, no hits yet
- 0x0000 = 48-byte TCAM hit on Entry 0x00[1]
- 0x0201 = 96-byte TCAM hit on Entries 0x01 (low) and 0x02 (high)
- 0x0003 = 48-byte TCAM hit on Entry 0x03
- 0x0809 = 96-byte TCAM hit on Entries 0x09 (low) and 0x08 (high)
- 0x0009 = 48-byte TCAM hit on Entry 0x09 with a TCAM miss on the high 48-bytes[2]
- 0x00FF = Miss on 1st 48 byte lookup

Note: A TCAM 'hit' on comparisons to frame bytes 1 to 48 are 'partial hits' if the selected TCAM entry (the lowest entry that matched) has the Continue bit set to a one. But these 'partial hits' are still recorded in the lower 8 bits of the TCAM Debug Register (above) so it can aide in debugging.

Note: Final TCAM 'hits' occur only if the last TCAM comparison results in a 'hit' to an entry whose Continue bits is cleared to a zero. Only this last entry's Action bits will be used on the frame.

1. This is a 'complete hit' if the selected TCAM entry has the Continue bit cleared to zero.
2. Assuming TCAM Entry 0x09 has the Continue bit set to one, the High Hit register bits being equal to 0x00 indicates a final TCAM 'miss' as the comparision to frame bytes 49 to 96 did not result in any 'hits'.

2 EEPROM Programming Format

The device supports an optional external serial EEPROM device for programming its internal registers. The EEPROM data will be read in once after Reset is deasserted. It can be re-started at any address (see Global 2 offset 0x14).

The device supports 32K bit (24C32) to 512K bit (24C512) EEPROM devices (the size and type of the device is automatically determined).

No mater what device is attached, the EEPROM device is read and processed in the same way:

1. Start at EEPROM address 0x0000.
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 1 offset 0x00) generating an Interrupt (if enabled).
9. Done.

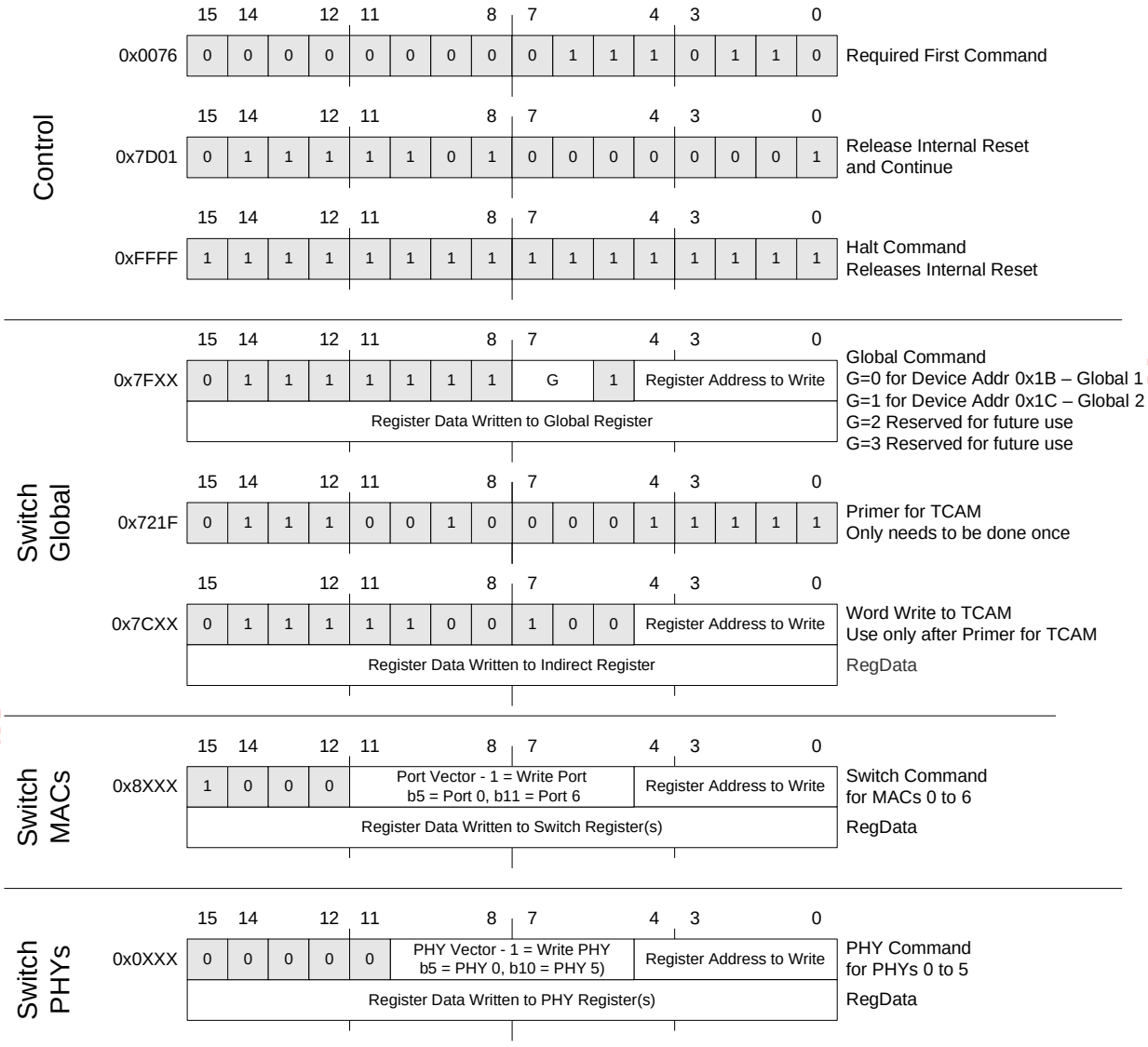
The 16-bit Command determines which register or registers inside the device are updated as follows (refer to Table 328 below):

1. In order for the EEPROM to process the commands defined in EEPROM Data Format (Rev 6), the first Command must be 0x0076 (0x76 must be at address 0x0000 & 0x00 must be at address 0x0001).
2. 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 bits 7:6, the G bits. If G=0 Global 1 at Device Addr 0x1B is written to. If G=1 Global 2 at Device Addr 0x1C is written to. All other values of the G bits are reserved for future use. Bits 4 to 0 determine the register to load.
3. For devices that support a TCAM, If the upper 10 bits of the Command is 0x7C8 then the TCAM will be written to if a primer command has been executed first. Bits 4 to 0 determine the register to load. The Primer for TCAM Command (0x721F) only needs to be executed once, but it must be executed before any Commands to write to TCAM.
4. Bit 15=1 the Switch MAC registers can be written to (SMI Device Addresses 0x10 to 0x16). A Device Vector (Command bits 11 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 MACs or to just some of them. Bits 4 to 0 determine the register to load.
5. When the upper 3 bits of the Command is 0x0, the integrated PHY registers can be written to (Internal SMI Device Addresses 0x10 to 0x15). A Device Vector (Command bits 10 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 PHYs or to just some of them. Bits 4 to 0 determine the register to load.
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 architecture minimizes the number of Command/RegData pairs that need to be stored in an EEPROM as only those registers that need to be changed from their defaults need to be placed in the EEPROM.

Table 328: EEPROM Data Format



Insert a Release Internal Reset command (0x7D01) whenever a Busy bit is set. This delays the next instruction until the Busy bit is cleared.

Marvell Technology Group
http://www.marvell.com

Marvell. Moving Forward Faster