ET3028-50
Single-Chip 28 x 1 Gbit/s Layer 2+ Ethernet Switch

Features

- Twenty-eight 10/100/1000 Mbits/s Ethernet ports:
  - — Twenty-four-ports supporting 6-pin SGMII interfaces to GbE copper PHYs
  - — Four SerDes ports
- Aggregate 42 Mpackets/s switching capacity (wire speed operation)
- 32-bit, 66 MHz *PCI* ™ processor interface
- One MDIO interface
- Integrated packet buffer memory
- Integrated address table memories:
  - — 8192 Layer 2 MAC addresses
- Full *IEEE*® *802.1d*® bridging
- Extensive VLAN support:
  - — Port-based VLANs
  - — Port-/protocol-based VLANs
  - — Up to 256 active VLANs
- L2/L3/L4 classification for access control list (ACL) and quality of service (QoS)
- Advanced traffic management functions:
  - — *802.3x* flow control
  - — Traffic shaping and scheduling
  - — Traffic policing
  - — Broadcast/multicast storm control
  - — Eight queues per port
- Link aggregation and mirroring
- Advanced 0.13 μm CMOS technology
- 525-pin FCBGA package

Benefits

- True switch-on-a-chip technology enables system vendors to build competitive 28-port gigabit Ethernet switches
- Integrated memories and SerDes for lower system power and cost
- Comprehensive QoS features and wire speed performance supporting enterprise desktop aggregation switching application
- Seamless interface and common API with Agere *TruePHY* ™ multiport PHY

Target Applications

- Layer 2 gigabit Ethernet desktop switches
- 28-port gigabit Ethernet switch in stand-alone configuration
- High-density gigabit Ethernet fabric switches

Block Diagram

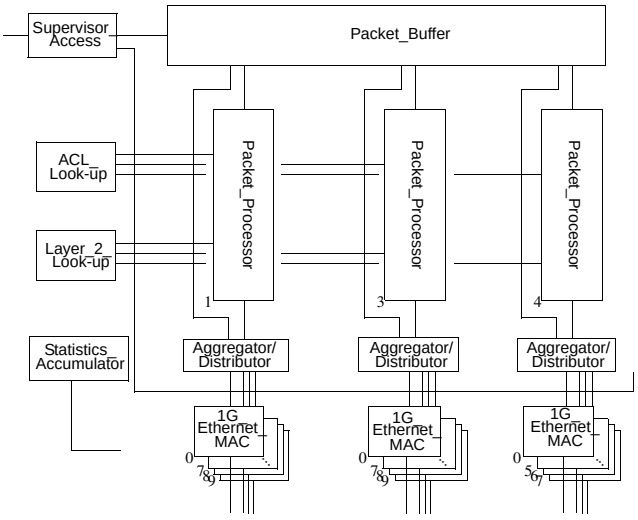


Figure 1. Block Diagram

Description

The ET3028-50 is a highly integrated and fully featured Layer 2 Ethernet bridge with integrated MACs, packet buffers, and address tables. The ET3028-50 contains 28 *IEEE 802.3z* 10/100/1000 Mbits/s Ethernet MACs. Twenty-four of the MACs connect to external PHYs via 6-pin LVDS SGMII interfaces. The remaining four MACs connect to external fiber PHYs via SerDes interfaces.

The ET3028-50 features an integrated L2/L3/L4 packet classification engine. The result of each packet classification is used to determine the packet's quality of service treatment and access control. Access to the network by individual desktops can be controlled by L2 MAC address and TCP/IP layer provisioning. Each port supports up to eight traffic class queues. Each packet is assigned to a class queue based on its *IEEE 802.1p* coding or IP TOS/DSCP value. Incoming traffic is policed to ensure the optimum use of network resources. Outgoing traffic is scheduled or shaped according to the traffic class and network resource usage.

Integrated address tables and packet buffers enable full bandwidth bridging on all ports at any legitimate frame length. All internal operations are at wire speed.

A reference system design kit is available for the ET3028-50. For more detailed product and reference design information, please contact Agere's local sales office.

System Diagram

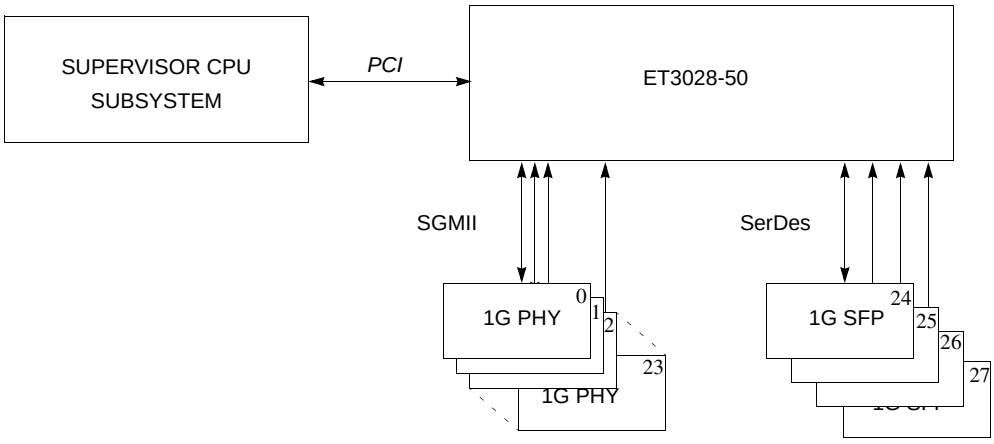


Figure 2. System Diagram

Table of Contents

Contents

Page

Table of Contents (continued)

Contents **Page**

Table of Contents (continued)

Contents **Page**

Table of Contents (continued)

Contents **Page**

Pin Descriptions

The ET3028-50 pins are summarized in the following series of tables. Each table is dedicated to a single interface type.

Table 1. 1G Ethernet SGMII Pins

Pin Name	Type	Pin Numbers	Description
RXCLK(23:00)_(P/N)	LVDS input	23{Y7,Y6}, 22{AA6,AA5}, 21{AB5,AB4}, 20{AA1,Y1}, 19{V4,W4}, 18{V3,U3}, 17{U4,T4}, 16{U1,T1}, 15{M6,N6}, 14{N4,M4}, 13{M3,L3}, 12{K1,J1}, 11{G6,H6}, 10{H4,G4}, 09{G3,F3}, 08{E1,D1}, 07{C4,C5}, 06{A4,A5}, 05{D5,D6}, 04{C6,C7}, 03{F9,F10}, 02{D10,D11}, 01{A11,A12}, 00{C11,C12}	SGMII receive clock differential pairs.
RXSD(23:00)_(P/N)	LVDS input	23{W7,W6}, 22{AA4,AA3}, 21{AB3,AB2}, 20{W1,V1}, 19{U5,V5}, 18{T3,R3}, 17{R4,P4}, 16{R1,P1}, 15{R6,P6}, 14{L4,K4}, 13{K3,J3}, 12{H1,G1}, 11{J6,K6}, 10{F4,E4}, 09{E3,D3}, 08{C1,B1}, 07{B3,B4}, 06{A6,A7}, 05{D7,D8}, 04{C8,C9}, 03{F7,F8}, 02{D12,D13}, 01{A13,A14}, 00{C13,C14}	SGMII receive data differential pairs.
TXSD(23:00)_(P/N)	LVDS output	23{V8,V7}, 22{Y5,Y4}, 21{AC3,AC2}, 20{Y2,W2}, 19{T6,U6}, 18{U2,V2}, 17{R5,T5}, 16{R2,T2}, 15{N5,P5}, 14{L5,M5}, 13{K2,L2}, 12{H2,J2}, 11{H5,J5}, 10{F5,G5}, 09{F2,E2}, 08{D2,C2}, 07{A2,A3}, 06{B5,B6}, 05{E6,E7}, 04{B7,B8}, 03{E8,E9}, 02{E11,E10}, 01{B13,B12}, 00{B15,B14}	SGMII transmit data differential pairs.
REFCLK_(2:0)	CMOS input	AC6, M1, A9	Reference clock for SGMII ports. Each should be set to 25 MHz.

Table 2. 1 Gbit/s Ethernet SFP Pins

Pin Name	Type	Pin #s	Description
RXSD(27:24)_(P,N)	CML input	27{AB12,AB13}, 26{Y9,Y10}, 25{W10,W11}, 24{V12,V13}	SFP SerDes receive differential pair.
TXSD(27:24)_(P,N)	CML output	27{Y13,Y14}, 26{AC11,AC12}, 25{AA10,AA11}, 24{V9,V10}	SFP SerDes transmit differential pair.
REFCLK_3_(P,N)	CML input	{AC9, AC8}	SFP SerDes differential clock input—LVDS or LVPECL levels compatible. 125 MHz nominal.
RESP_3	REF	AB9	SFP SerDes reference resistor connection. Tie to VSSRESP_3 through a 1.5 kΩ resistor.
VSSRESP_3	REF	AB8	GND connection for RESP_3.

Pin Descriptions (continued)

Table 3. Supervisor *PCI* Pins

Pin Name	Type	Pin Numbers	Description
PCI_RST_N	CMOS input	P19	*PC*I reset. Active-low.
PCI_CLK	CMOS input	AC20	*PCI* bus clock.
AD(31:00)	CMOS I/O	31{P22}, 30{T20}, 29{R18}, 28{R22}, 27{T23}, 26{T18}, 25{R19}, 24{U23}, 23{T21}, 22{U19}, 21{T19}, 20{V19}, 19{W19}, 18{V18}, 17{V23}, 16{V21}, 15{AA21}, 14{AB22}, 13{AA23}, 12{AB23}, 11{AC22}, 10{AB21}, 09{AC21}, 08{AB20}, 07{AA20}, 06{W16}, 05{AC19}, 04{V16}, 03{AA19}, 02{Y18}, 01{AB19}, 00{AA18},	*PCI* address and data bus.
CBE(3:0)_N	CMOS I/O	3{T22}, 2{V22}, 1{AA22}, 0{P21}	*PCI* command and byte-enable signals. Active-low.
PAR	CMOS I/O	Y20	*PCI* parity signal.
FRAME_N	CMOS I/O	V17	*PCI* cycle frame signal. Active-low.
IRDY_N	CMOS I/O	W22	*PCI* initiator ready signal. Active-low.
TRDY_N	CMOS I/O	W21	*PCI* target ready signal. Active-low.
DEVSEL_N	CMOS I/O	W18	*PCI* device select signal. Active-low.
STOP_N	CMOS I/O	W20	*PCI* stop signal. Active-low.
PERR_N	CMOS I/O	W23	*PCI* parity error signal. Active-low.
SERR_N	CMOS I/O	Y22	*PCI* system error signal. Active-low.
IDSEL	CMOS input	U21	*PCI* initialization device select signal.
REQ_N	CMOS output	R23	*PCI* bus request signal. Active-low.
GNT_N	CMOS input	R21	*PCI* bus grant signal. Active-low.
INTA_N	CMOS output (open drain)	R20	*PCI* interrupt. Active-low, open-drain.
LOCK_N	CMOS input	V20	*PCI* lock signal. Active-low.

Pin Descriptions (continued)

Table 4. Miscellaneous Pins

Pin Name	Type	Pin Numbers	Description
TDI	CMOS input	C15	JTAG serial data in.
TCK	CMOS input	D15	JTAG clock.
TMS	CMOS input	E14	JTAG test mode select.
TRST_N	CMOS input	E13	JTAG test reset. Active-low. Pull low for normal operation.
TDO	CMOS output	F12	JTAG serial data out.
MDIO_C22	CMOS I/O (open drain)	AA16	Serial management data I/O signal. Clause 22 compatible.
MDC_C22	CMOS output	AA17	Serial management clock signal. Clause 22 compatible.
SE	CMOS input	AB18	Reserved—scan enable. Pull low for normal operation.
TM_(2:0)	CMOS input	2{F13}, 1{F14}, 0{F15}	Reserved—test mode select. Pull low for normal operation.
TST_(9:0)	CMOS output	9{U17}, 8{Y17}, 7{AB15}, 6{Y16}, 5{U16}, 4{W15}, 3{T16}, 2{V15}, 1{T17}, 0{U15}	Reserved—test mode output. Leave unconnected.
VM _(1:0)	CMOS output	1{P20}, 0{P18}	Reserved—voltage monitor output. Leave unconnected.
RREF_(2:0)	REF	2{AB6}, 1{N2}, 0{B9}	Bias reference—pull to ground with individual 2.49 kΩ resistors.
VREF	REF	H7	Reference voltage—set to 1.23 V ± 1%.
RST_N	CMOS input	P23	Chip reset, active-low. During powerup, this signal should be held low for at least 200 ns after power has ramped and REFCLK_CORE (25 MHz) is stable. If the system is already powered up and stable, the signal may be pulsed low for 200ns or more. See Device Reset in Appendix B.
REFCLK_CORE	CMOS input	AC16	Reference clock for the core of the chip. 25 MHz.
NO_CONNECT	NC	AB17, AC18, N20, N21, L23, H23, G23, M23, K21, L21, K19, L19, M18, N18, F19, G19, F21, G21, H18, J18, H20, J20, D20, E20, B21, C21, D22, E22, B23, C23, E16, E17, A16, A17, C18, C19, A19, A20	Not used—leave unconnected.

Pin Descriptions (continued)

Table 5. Power and Ground

Pin Name	Type	Pin Numbers	Description
VDD12_CORE	PWR	G8, G10, G12, G14, G16, H9, H11, H13, H15, H17, J8, J10, J12, J14, J16, K7, K9, K11, K13, K15, K17, L8, L10, L12, L14, L16, M7, M9, M11, M13, M15, M17, N8, N10, N12, N14, N16, P7, P9, P11, P13, P15, P17, R8, R10, R12, R14, R16, T7, T9, T11, T13, T15, U8, U10, U12, U14	Core power supply. 1.2 V, nominal.
VSS_CORE	GND	B11, B16, C3, D9, E5, E12, F1, G7, G9, G11, G13, G15, G17, H3, H8, H10, H12, H14, H16, J7, J9, J11, J13, J15, J17, K5, K8, K10, K12, K14, K16, L7, L9, L11, L13, L15, L17, M8, M10, M12, M14, M16, N3, N7, N9, N11, N13, N15, N17, P2, P8, P10, P12, P14, P16, R7, R9, R11, R13, R15, R17, T8, T10, T12, T14, U7, U9, U11, U20, W3, W5, W17, Y21, AB1, AB16	Core ground.
VDD15	PWR	G20, G22, J19, J21; L22, K20, M19, N19, F17, B17, D18, B20; C20, E21, C22, B22	Power supply. 1.5 V, nominal.
VDD15L	PWR	Y8, AC4, Y3, V6, L6, J4, B2, D4, B10, F11, A15, D16	Analog power supply for SGMII I/O. 1.5 V, nominal.
VDD33	PWR	E15, Y23, Y19, U22, U18, AA15	Power supply. 3.3 V, nominal.
VDD33L	PWR	C10, D14, F6, G2, M2, P3, AA2, AA7	Power supply. SGMII I/O. 3.3 V, nominal.
AVDD12_PLL	PWR	AC7, N1, A8	Analog power supply for SGMII PLLs. 1.2 V, nominal.
AGND_PLL	GND	AC5, L1, A10	Analog ground for SGMII PLLs.
AVDD12_PLL_CORE	PWR	AC15	Analog power supply for core PLLs. 1.2 V, nominal.
AGND_PLL_CORE	GND	AC17	Analog ground for core PLLs.
VDDIB_(27:24)	PWR	27{AA13}, 26{AA8}, 25{Y12}, 24{V11}	Analog input buffer power supply. 1.5 V, nominal.
VDDOB_(27:24)	PWR	27{AA12}, 26{AC13}, 25{AB10}, 24{W9}	Analog output buffer power supply. 1.5 V, nominal.
VDD_3	PWR	AA9, AA14, AB7, AC14, W8, W12, W14	SFP power supply. 1.2 V, nominal.
VDD_4	PWR	A22, B18, C16, D17, D19, D23, E18, F20, F23, G18, H19, H22, J22, K23, M20, M22	Power supply. 1.2 V, nominal.
VSS_3	GND	AB11, AB14, AC10, U13, V14, W13, Y11, Y15	SFP ground.
VSS_4	GND	A18, A21, B19, C17, D21, E19, E23, F16, F18, F22, H21, J23, K18, K22, L18, L20, M21, N22, N23	Ground.

Memory Map

The following table lists all of the ET3028-50's registers in the order they appear in the address space allocated to the device. The base addresses listed below must be added to the base address configured for the device in order to arrive at the actual physical address of each register.

Table 6. Memory Map

Base Address	Size	Register Name
0x0004_4000	2048	Rx_Length_Histogram
0x0004_4800	2048	Tx_Length_Histogram
0x0004_5000	1024	Tx_Collision_Histogram
0x0004_5400	256	Tx_Total_Collisions
0x0004_5800	1024	Rx_Error_Packets
0x0004_5c00	1024	Rx_Packets
0x0004_6000	1024	Tx_Packets
0x0004_6400	256	Rx_Bytes
0x0004_6500	256	Tx_Bytes
0x0004_8000	16384	Vlan_Index_Table_0
0x0004_c000	4096	Policer_Flow_Id_Table_0
0x0004_d000	1024	Vlan_Id_Table_0
0x0004_d400	512	Policer_Accumulator_Table_0
0x0004_d600	512	Policer_Delta_Table_0
0x0004_d800	512	Policer_Flow_Mode_Table_0
0x0004_da00	512	Policer_Limit_Table_0
0x0004_dc00	512	Vlan_Port_Protocol_Table_0
0x0004_de00	256	Priority_Encode_Table_0
0x0004_df00	128	Priority_Decode_Table_0
0x0004_df80	64	Port_Mode_0
0x0004_dfc0	16	User_Protocol_0
0x0004_dfd0	16	User_Type_0
0x0004_dfe0	4	Policer_Mode_0
0x0004_e000	1024	Policer_Statistics_0
0x0005_0000	16384	Vlan_Index_Table_1
0x0005_4000	4096	Policer_Flow_Id_Table_1
0x0005_5000	1024	Vlan_Id_Table_1
0x0005_5400	512	Policer_Accumulator_Table_1
0x0005_5600	512	Policer_Delta_Table_1
0x0005_5800	512	Policer_Flow_Mode_Table_1
0x0005_5a00	512	Policer_Limit_Table_1
0x0005_5c00	512	Vlan_Port_Protocol_Table_1
0x0005_5e00	256	Priority_Encode_Table_1
0x0005_5f00	128	Priority_Decode_Table_1
0x0005_5f80	64	Port_Mode_1
0x0005_5fc0	16	User_Protocol_1
0x0005_5fd0	16	User_Type_1
0x0005_5fe0	4	Policer_Mode_1
0x0005_6000	1024	Policer_Statistics_1
0x0005_8000	16384	Vlan_Index_Table_2

Memory Map (continued)

Table 6. Memory Map (continued)

Base Address	Size	Register Name
0x0005_c000	4096	Policer_Flow_Id_Table_2
0x0005_d000	1024	Vlan_Id_Table_2
0x0005_d400	512	Policer_Accumulator_Table_2
0x0005_d600	512	Policer_Delta_Table_2
0x0005_d800	512	Policer_Flow_Mode_Table_2
0x0005_da00	512	Policer_Limit_Table_2
0x0005_dc00	512	Vlan_Port_Protocol_Table_2
0x0005_de00	256	Priority_Encode_Table_2
0x0005_df00	128	Priority_Decode_Table_2
0x0005_df80	64	Port_Mode_2
0x0005_dfc0	16	User_Protocol_2
0x0005_dfd0	16	User_Type_2
0x0005_dfe0	4	Policer_Mode_2
0x0005_e000	1024	Policer_Statistics_2
0x0006_0000	16384	Vlan_Index_Table_3
0x0006_4000	4096	Policer_Flow_Id_Table_3
0x0006_5000	1024	Vlan_Id_Table_3
0x0006_5400	512	Policer_Accumulator_Table_3
0x0006_5600	512	Policer_Delta_Table_3
0x0006_5800	512	Policer_Flow_Mode_Table_3
0x0006_5a00	512	Policer_Limit_Table_3
0x0006_5c00	512	Vlan_Port_Protocol_Table_3
0x0006_5e00	256	Priority_Encode_Table_3
0x0006_5f00	128	Priority_Decode_Table_3
0x0006_5f80	64	Port_Mode_3
0x0006_5fc0	16	User_Protocol_3
0x0006_5fd0	16	User_Type_3
0x0006_5fe0	4	Policer_Mode_3
0x0006_6000	1024	Policer_Statistics_3
0x0006_8000	16384	Vlan_Index_Table_4
0x0006_c000	4096	Policer_Flow_Id_Table_4
0x0006_d000	1024	Vlan_Id_Table_4
0x0006_d400	512	Policer_Accumulator_Table_4
0x0006_d600	512	Policer_Delta_Table_4
0x0006_d800	512	Policer_Flow_Mode_Table_4
0x0006_da00	512	Policer_Limit_Table_4
0x0006_dc00	512	Vlan_Port_Protocol_Table_4
0x0006_de00	256	Priority_Encode_Table_4
0x0006_df00	128	Priority_Decode_Table_4
0x0006_df80	64	Port_Mode_4
0x0006_dfc0	16	User_Protocol_4
0x0006_dfd0	16	User_Type_4
0x0006_dfe0	4	Policer_Mode_4

Memory Map (continued)

Table 6. Memory Map (continued)

Base Address	Size	Register Name
0x0006_e000	1024	Policer_Statistics_4
0x0008_0000	131072	Layer_2_Key_Table_6
0x000a_0000	65536	Layer_2_Vlan_Port_State_Table
0x000b_0000	32768	Layer_2_Key_Table_5
0x000b_8000	32768	Layer_2_Time_Stamp_Table
0x000c_0000	8192	Layer_2_Key_Table_4
0x000c_2000	4096	Layer_2_Dest_Map_Table
0x000c_3000	2048	Layer_2_Key_Table_3
0x000c_3800	2048	Layer_2_Vlan_Mask_Table
0x000c_4000	512	Layer_2_Key_Table_2
0x000c_4200	512	Layer_2_Src_Port_Mask_Table
0x000c_4400	128	Layer_2_Key_Table_1
0x000c_4480	64	Layer_2_Aggregation_Mask_Table
0x000c_4500	256	Layer_2_Logical_Port_Table
0x000c_4600	32	Layer_2_Key_Table_0
0x000c_4620	8	Layer_2_Active_Port_Map
0x000c_4628	8	Layer_2_Blocking_Mask
0x000c_4630	8	Layer_2_Dest_Mirror_Map
0x000c_4638	8	Layer_2_Flood_Map
0x000c_4640	8	Layer_2_Global_Mask
0x000c_4648	8	Layer_2_Learning_Mask
0x000c_4650	8	Layer_2_Src_Deny_Mask
0x000c_4658	8	Layer_2_Src_Mirror_Map
0x000c_4660	4	Layer_2_Current_Time
0x000c_4664	4	Layer_2_Igmp_Snooping_Port
0x000c_4668	4	Layer_2_Learning_Port
0x000c_466c	4	Layer_2_Mirror_Port
0x000c_4674	4	Layer_2_Supervisor_Route_Port
0x000c_4678	4	Layer_2_User_Port
0x000c_467c	4	Layer_2_User_Port_Snooping_Port
0x000c_4680	4	Layer_2_No_Dest_Packets
0x000c_4700	244	Layer_2_Mode
0x000c_4800	8	Multicast_Rate_Discard_Mask
0x000c_4808	4	Multicast_Rate_Limit
0x000c_480c	4	Multicast_Rate_Decrement_Period
0x000c_4810	4	Multicast_Rate_Limit_Events
0x000c_4814	4	Multicast_Rate_Accumulator
0x000c_4818	4	Multicast_Rate_Mode
0x000c_8000	64	Mac_Mode_0
0x000c_8050	4	Mac_Status_0
0x000c_8080	64	Mac_Mode_1
0x000c_80d0	4	Mac_Status_1
0x000c_8100	64	Mac_Mode_2

Memory Map (continued)

Table 6. Memory Map (continued)

Base Address	Size	Register Name
0x000c_8150	4	Mac_Status_2
0x000c_8180	64	Mac_Mode_3
0x000c_81d0	4	Mac_Status_3
0x000c_8200	64	Mac_Mode_4
0x000c_8240	12	Serdes_Control_4
0x000c_8250	4	Mac_Status_4
0x000c_8300	4	Mdio_Control
0x000c_8304	4	Mdio_Status
0x000c_8308	4	Mdio_Mode
0x000c_8310	16	Mac_Global_Mode
0x000c_8320	4	Device_Version
0x000c_a000	2048	Packet_Buffer_Queue_En
0x000c_a800	1024	Packet_Buffer_Shaper_Accumulator_Even
0x000c_ac00	1024	Packet_Buffer_Shaper_Accumulator_Odd
0x000c_b000	1024	Packet_Buffer_Shaper_Delta_Even
0x000c_b400	1024	Packet_Buffer_Shaper_Delta_Odd
0x000c_b800	256	Packet_Buffer_Channel_Congestion_Threshold
0x000c_b900	256	Packet_Buffer_Allocated_Buffer_Count
0x000c_ba00	256	Packet_Buffer_Queue_Status
0x000c_bb00	64	Packet_Buffer_Packet_Drop_Count
0x000c_bb40	32	Packet_Buffer_Ind
0x000c_bb60	16	Packet_Buffer_Descriptor_Usage_Count
0x000c_bb70	4	Packet_Buffer_Free_Buffer_Control
0x000c_bb74	4	Packet_Buffer_Free_Descriptor_Control
0x000c_bb78	4	Packet_Buffer_Parity_Error_Info
0x000c_bb80	4	Packet_Buffer_Scrub
0x000c_bb84	4	Packet_Buffer_Free_Buffer_Count_Even
0x000c_bb88	4	Packet_Buffer_Free_Buffer_Count_Odd
0x000c_bc00	64	Packet_Buffer_Priority_Table
0x000c_bc48	8	Packet_Buffer_Crc_Error_Mask
0x000c_bc50	8	Packet_Buffer_Discard_Mask
0x000c_bc5c	4	Packet_Buffer_Descriptor_Congestion_Threshold
0x000c_bc60	4	Packet_Buffer_Global_Congestion_Threshold
0x000c_bc64	4	Packet_Buffer_Mode
0x000c_bc68	4	Packet_Buffer_Shaper_Limit
0x000c_bd00	192	Packet_Buffer_Port_Speed
0x000c_c400	64	Supervisor_Rx_Fifo_Ptr
0x000c_c440	8	Supervisor_Tx_Fifo_Ptr
0x000c_c448	4	Supervisor_Ind
0x000c_c44c	4	Supervisor_Ind_Mask
0x000c_c450	4	Supervisor_Ind_Mask_Clear
0x000c_c454	4	Supervisor_Ind_Mask_Set
0x000c_c458	4	Supervisor_Int

Memory Map (continued)

Table 6. Memory Map (continued)

Base Address	Size	Register Name
0x000c_c45c	4	Supervisor_Int_Mask
0x000c_c460	4	Supervisor_Int_Mask_Clear
0x000c_c464	4	Supervisor_Int_Mask_Set
0x000c_c468	4	Supervisor_Invalid_Addr
0x000c_c46c	4	Supervisor_Statistics_Transfer_Addr
0x000c_c470	4	Supervisor_Statistics_Transfer_Status
0x000c_c480	128	Supervisor_Rx_Fifo_Limits
0x000c_c500	32	Supervisor_Tx_Fifo_Limits
0x000c_c520	4	Supervisor_Endian
0x000c_c540	32	Supervisor_Maximum_Packet_Length
0x000c_d000	428	Packet_Buffer_Queue_Depth_0 (revision C only)
0x000c_d200	428	Packet_Buffer_Queue_Depth_1 (revision C only)
0x000c_d400	428	Packet_Buffer_Queue_Depth_2 (revision C only)
0x000c_d600	428	Packet_Buffer_Queue_Depth_3 (revision C only)
0x000c_da00	424	Packet_Buffer_Queue_Buffer_0 (revision C only)
0x000c_dc00	424	Packet_Buffer_Queue_Buffer_1 (revision C only)
0x000c_de00	424	Packet_Buffer_Queue_Buffer_2 (revision C only)
0x000c_e000	424	Packet_Buffer_Queue_Buffer_3 (revision C only)
0x000c_e200	4	Packet_Buffer_Queue_Buffer_Limit (revision C only)
0x000c_e204	4	Packet_Buffer_Queue_Limit (revision C only)
0x000c_e208	4	Packet_Buffer_Queue_Management_Thresholds (revision C only)

Functional Description

This chapter describes the packet handling processes embedded within the ET3028-50.

Packet Reception

The packet reception process quite simply receives packets from the attached network, immediately processes control packets, parses the remaining packets, and delivers the received packets to the next process.

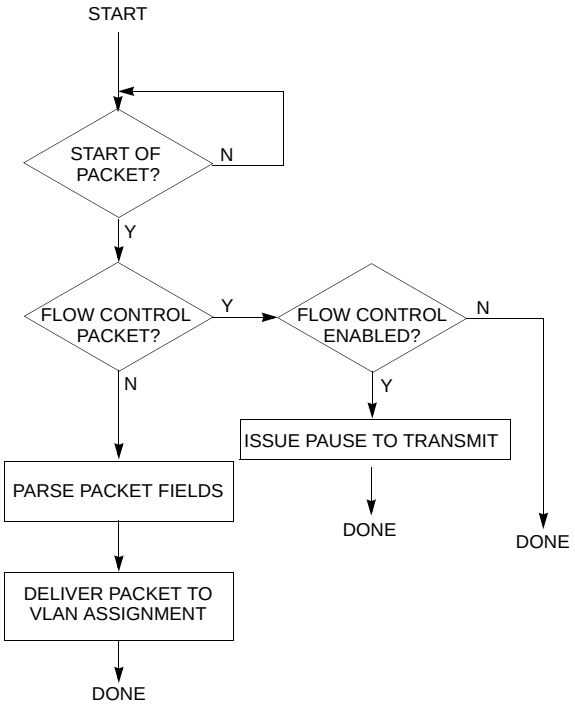


Figure 3. Packet Reception Process

Functional Description (continued)

Packet Reception (continued)

The packet reception process is rather straightforward. If a receive packet is a flow-control packet and flow control is enabled, an indication is provided to the MAC's corresponding transmit function that causes it to cease transmitting for a specified period of time. All other receive packets are parsed and then passed along to the VLAN assignment process.

The following figure shows the various packet formats that are supported by the ET3028-50's parser function.

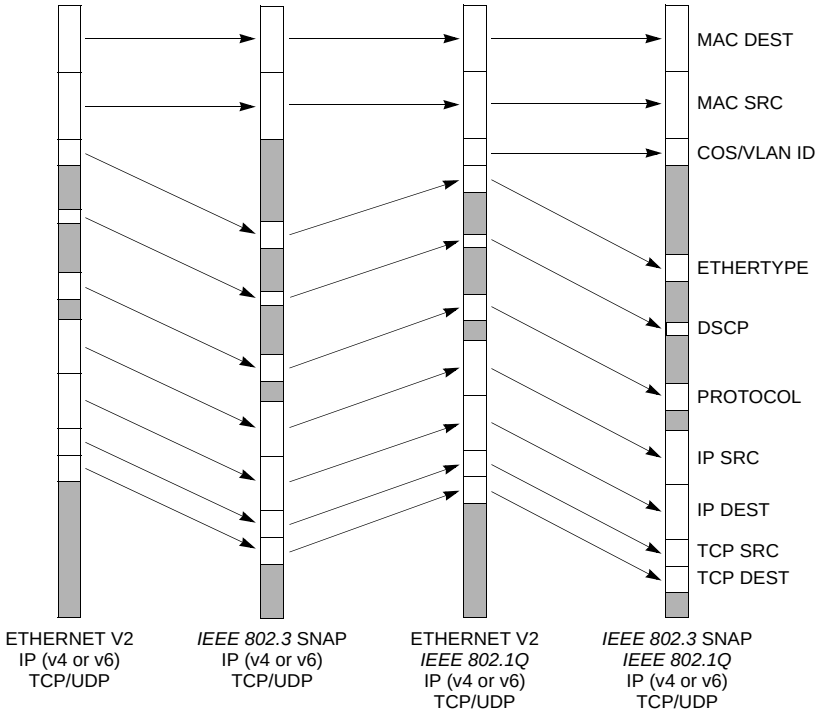


Figure 4. Packet Formats

The packet reception process automatically restarts after each packet.

Functional Description (continued)

VLAN Assignment

All packets received by the ET3028-50 are assigned to a VLAN. Some packets are received with a VLAN tag in place. For the remainder of the packets, other aspects of the packets must be used to determine to which VLAN the packet is to be assigned.

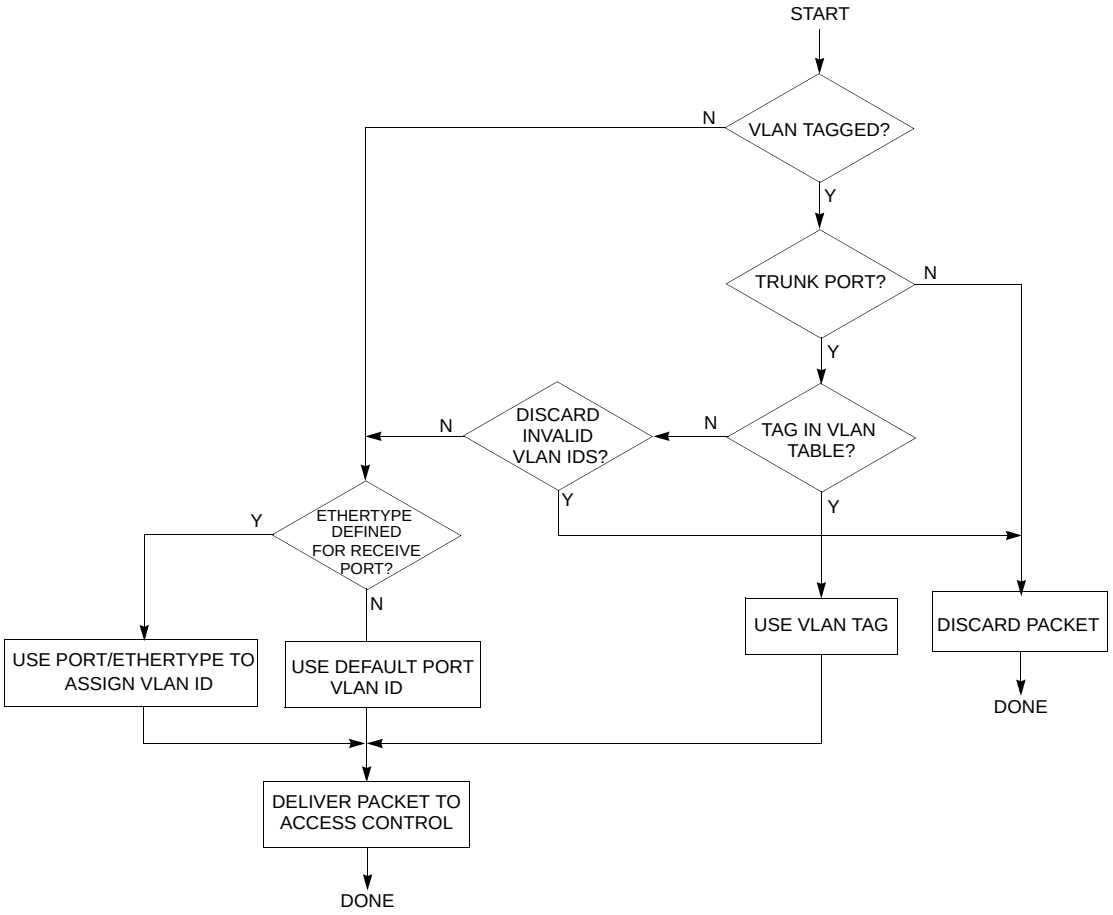


Figure 5. VLAN Assignment Process—Revisions B and B1

In revision B and B1, VLAN tagged packets may only be received by those Ethernet ports that are configured as trunk ports. If a tagged packet is received via an access port, it is discarded. In revision C, VLAN tagged packets may be received by either access-configured or trunk-configured Ethernet ports. The VLAN identifiers in tagged packets must also exist in the ET3028-50's VLAN index table. Packets with invalid VLAN tags may be optionally discarded. If packets with invalid VLAN tags are not discarded, they are treated as if they are untagged packets.

For untagged packets, an attempt is made to assign the packet to a VLAN based on its combination of receive port and ethertype. If the combination found in the receive packet is not present in the port/protocol tables, then the receive port's default VLAN identifier is used for the receive packet.

Upon completion of the VLAN assignment process, the packet is delivered to the access control process.

Functional Description (continued)

VLAN Assignment (continued)

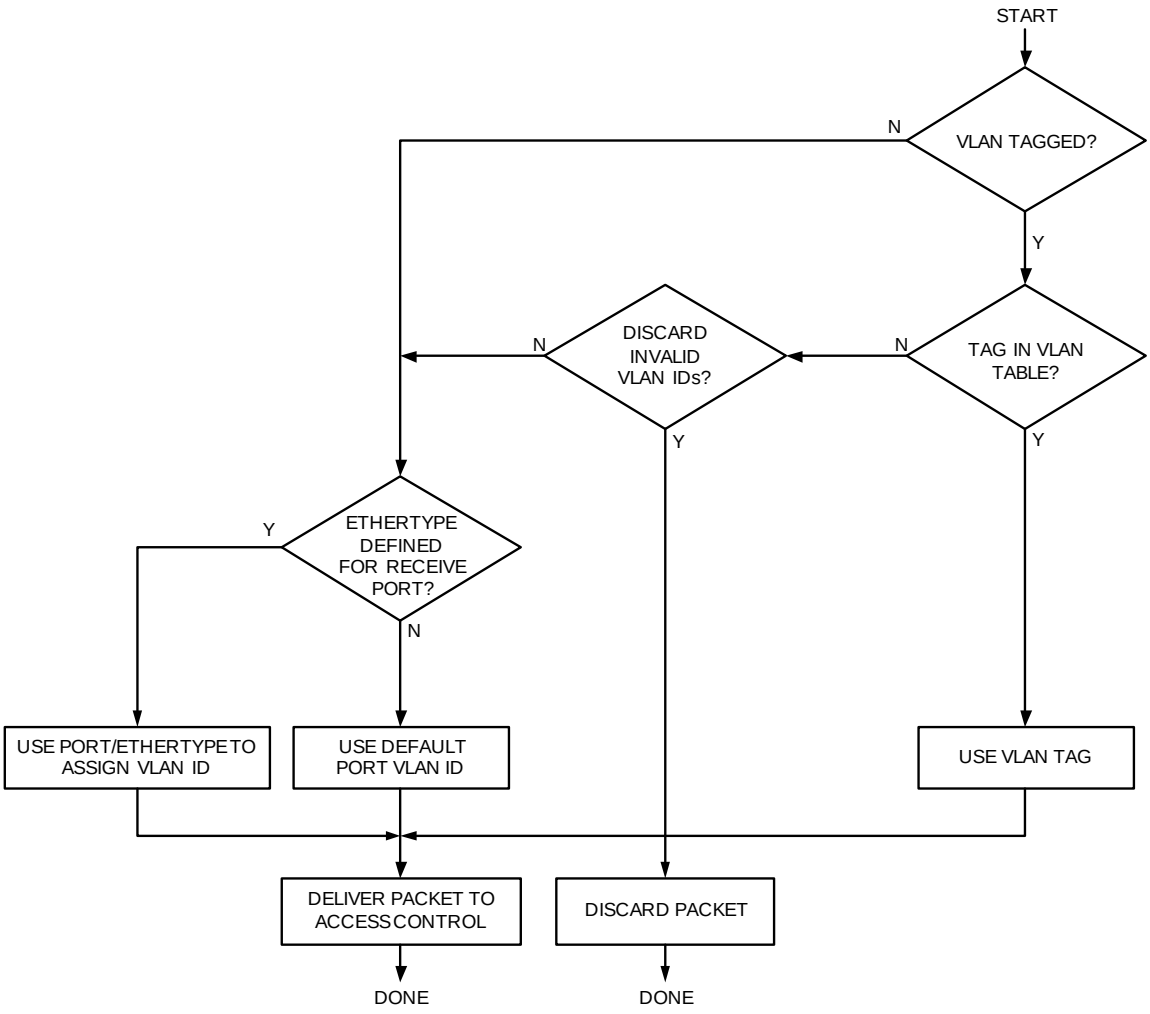


Figure 6. VLAN Assignment Process—Revision C

Functional Description (continued)

Access Control

The access control process examines certain fields within each receive packet and determines whether or not the packets should be granted access to the ET3028-50 and its attached networks.

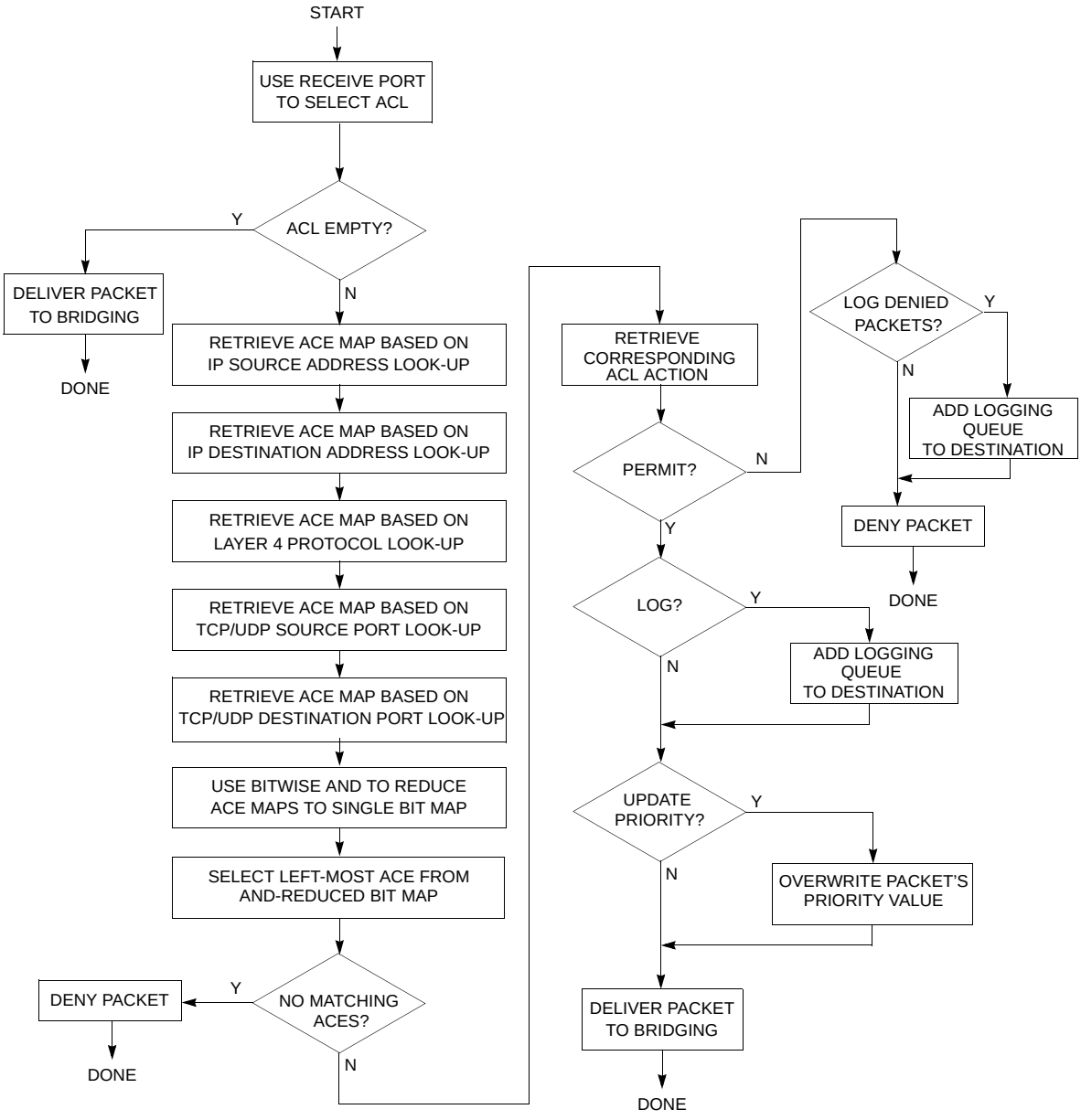


Figure 7. Access Control Process

Functional Description (continued)

Access Control (continued)

Receive packets are associated with access control lists (ACLs) based on their receive port. If the ACL associated with a receive packet is empty, then the packet is automatically permitted to be received and processed normally. No further action is taken by the ACL process in this case.

For a nonempty ACL, each of the packet's fields that make up the ACL 5-tuple are submitted to a look-up, which ultimately results in the gathering of five access control entry (ACE) map values. These ACE maps indicate which of the ACEs that make up the ACL have criteria that are satisfied by the corresponding field in the receive packet.

For example, a TCP destination port value of 3,027 may satisfy one rule that stipulates a range of 3,000 to 4,000 and another that specifies all values less than 7,000. In this example, 2 bits in the corresponding ACE map are asserted with each bit indicating which ACE has matching criteria.

Once all five ACE maps have been retrieved, they are ANDed together in a bitwise fashion to indicate those ACEs where all five criteria are satisfied by the fields in the receive packet. The leftmost bit in the resulting ACE map indicates the first ACE in the ACL that matches all of the criteria. In the case where multiple ACEs match the against the receive packet, it is the one that appears first in the ACL that is applied to the receive packet.

The actions that may be carried out as a result of a matching ACE are as follows:

1. Packet denial or permission
2. Packet logging
3. Priority reassignment

The ET3028-50 may also be configured to automatically log all denied packets.

Functional Description (continued)

Bridging

The ET3028-50's bridging process utilizes Layer 2 information from the packet as a primary means for determining the destination or destinations of a receive packet.

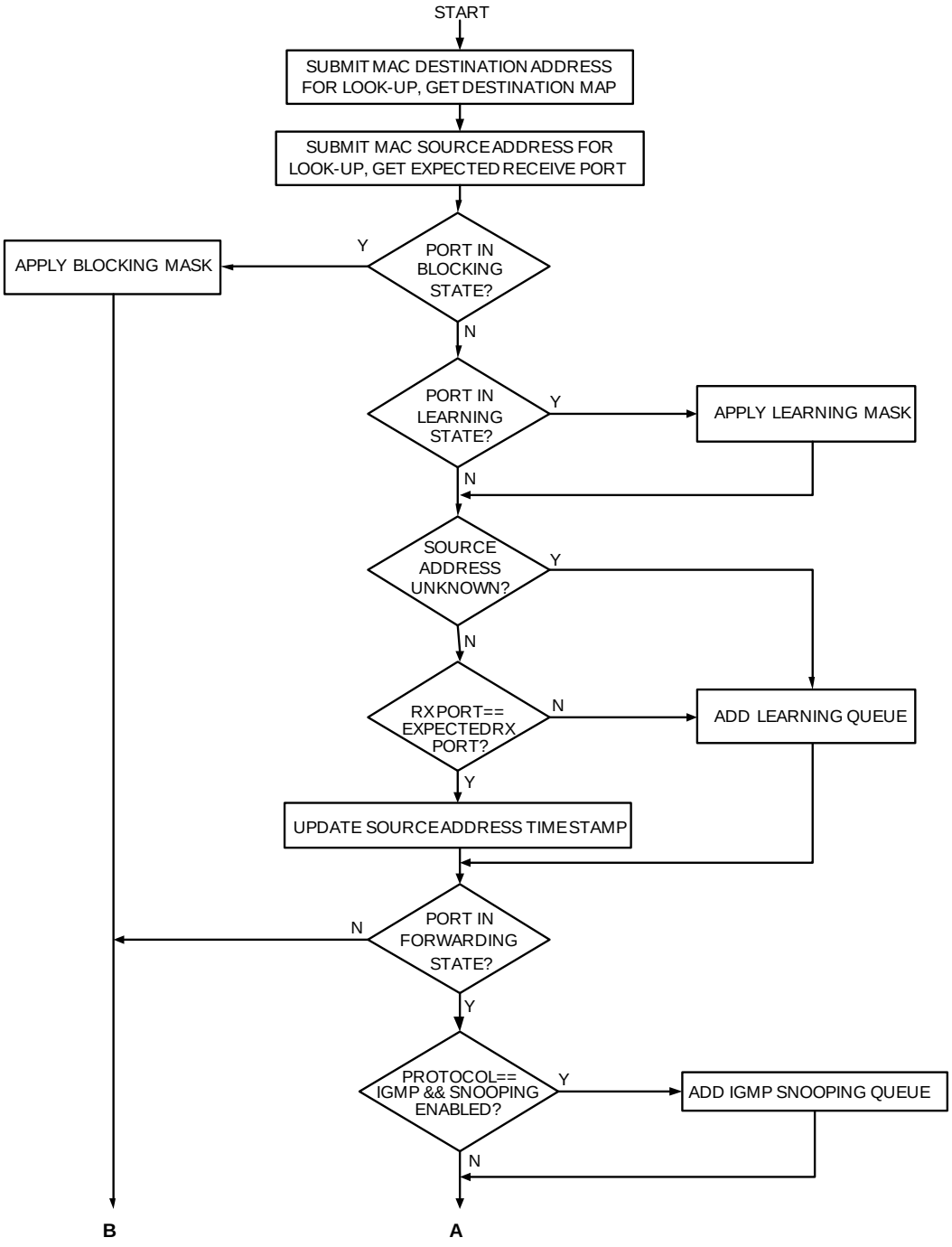


Figure 8. Bridging Process (Part 1)

Functional Description (continued)

Bridging (continued)

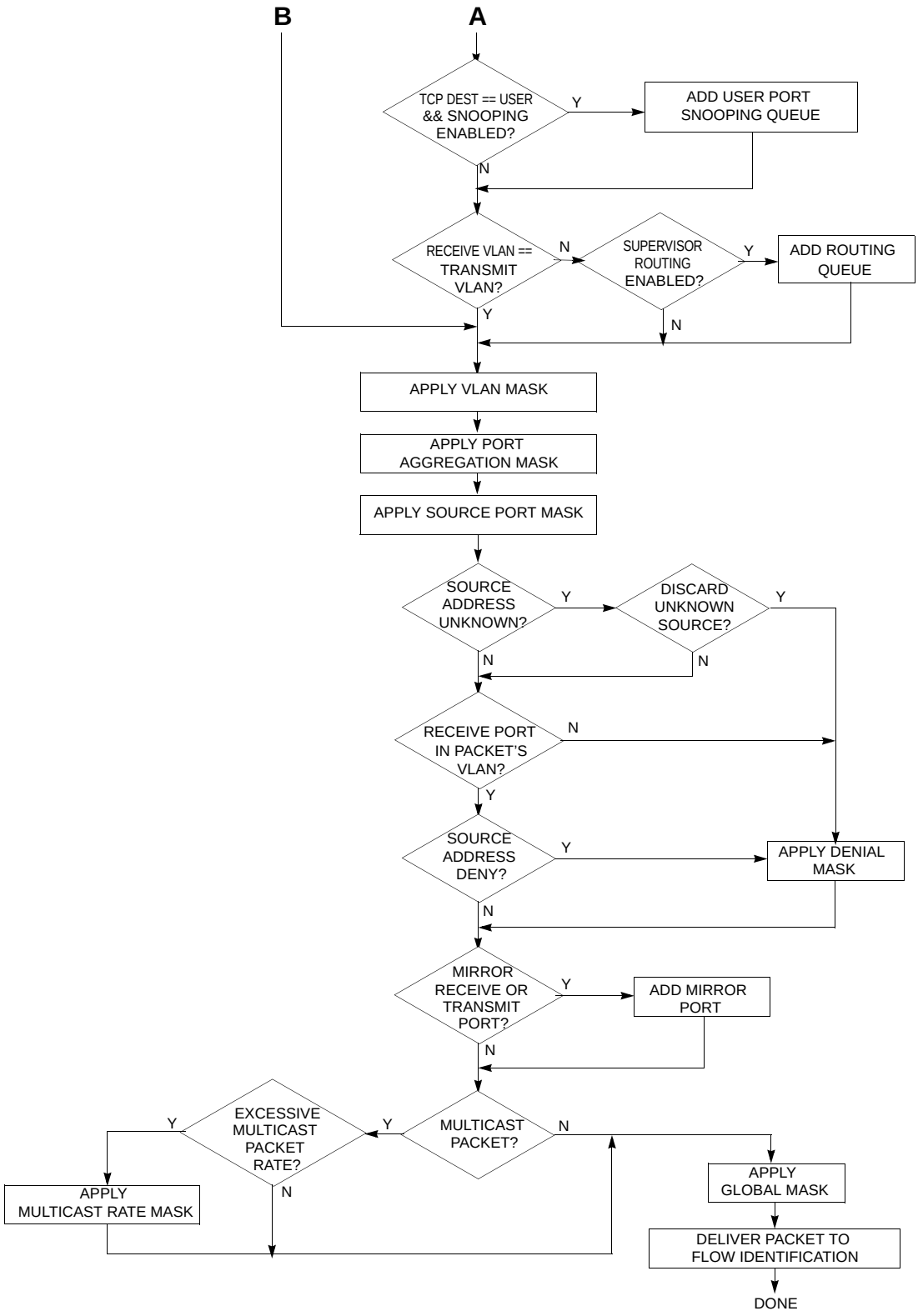


Figure 9. Bridging Process (Part 2)

Functional Description (continued)

Bridging (continued)

The bridging process starts with both the Layer 2 destination and source addresses being submitted for a look-up. The principal information returned is the destination map and the source port associated with the packet's source address. Other information returned includes the match/fail result status for each look-up, flow identifying tags, (used in the flow identification process described below) and a Layer 2 access control flag.

The destination map provides a per-port indication of which ports are destinations for the packet. An asserted bit in the map implies that the corresponding port is an intended destination. The look-up of the destination address returns an initial destination map. This map is then modified through the application of various maps (adding destinations) and masks (deleting destinations). The final resulting destination map is used in concert with the packet's priority information to identify the queues into which references to the receive packet are deposited.

First, the spanning tree state of the receive port is used to modify the packet's destination. The blocking mask disables all destinations that should be unreachable to a receive port that is in the blocking state.

Note: The ET3028-50 treats the blocking and listening states as being functionally equivalent.

The learning mask deletes those destinations that should be unreachable to a receive port that is in the learning state. If the receive packet's source address is unknown or its receive port does not match the receive port associated with the source address in the address table, the ET3028-50's learning queue is added to the packet's list of destinations.

Note: The link aggregation function makes it possible for a source address to appear on multiple receive ports within a very short time span. Ordinarily, these varying source ports would cause many unnecessary copies of the packet to be forwarded to the learning queue. The use of a logical port number that groups the ports associated with an aggregate together under a single label eliminates these unnecessary learning operations. Refer to Link Aggregation on page 240 for more information regarding the configuration of link aggregates.

If the source address of the receive packet does exist in the address table, and the stored receive port number matches the port number via which the packet was received, then the address table's time stamp value for the source address is updated. This is a simple matter of writing the current time value to the time stamp table location that corresponds to the current source address of the receive packet.

IGMP and user port snooping are performed next. For IGMP snooping, if the receive packet is an IGMP packet and IGMP snooping is enabled, then the IGMP snooping queue is added to the destination map. Similarly, if the TCP or UDP destination port number matches the user-defined snooping port number, then the user-port snooping queue is added to the packet's destination map.

If the VLAN associated with the receive packet does not match the VLANs associated with any of the destination ports indicated in the destination map so far, then the packet is not going to be forwarded due to VLAN filtering. However, if so enabled, the supervisor may receive these VLAN-filtered packet so that it may apply routing operations to these packets and allow them to communicate across VLAN boundaries. The supervisor's routing queue is used for this purpose.

The following three masks are applied unconditionally:

1. VLAN mask

2. Port aggregation mask

3. Source port mask

The VLAN mask is applied to delete all of the destinations in the destination map that are not members of the receive packet's VLAN.

Functional Description (continued)

Bridging (continued)

The port aggregation mask is one of eight masks that are applied to all destination maps, regardless of a packet's receive port or its intended destinations. It is presumed that all ports are a member of an aggregate. An aggregate may be as small as a single port. In this case, the single port in the aggregate would never be eliminated by any of the aggregate masks. For multiport aggregates, each aggregate mask eliminates all but one destination port for all of the aggregates configured in the device. For example, for a 4-port aggregate, two of the eight aggregate masks are configured to allow a particular port to remain in the destination map while the remaining six are configured to delete that same port.

The source port mask is used to disallow communication between a receive port and an arbitrary collection of transmit ports. This capability enables features such as private VLANs. A private VLAN exists when an access port and its related trunk are allowed to communicate with one another, but the various access ports associated with a trunk port are not allowed to communicate with one another. This has the effect of allowing only client/server communication and disallowing client/client communication.

Next, a series of denial tests are applied. If the source address is unknown and the port is configured to deny packets with unknown source addresses, the denial mask is applied to the destination map. If the receive port is not a member of the receive packet's VLAN, the denial mask is applied. Finally, if the source permit flag for the source address is false, the denial mask is applied to the destination map.

If port mirroring is enabled for either the receive port or any of the destination ports, then the designated mirror port is added to the packet's destination map.

If the receive packet is a multicast packet and the rate at which multicast packets are being received exceeds a configured limit, then a multicast rate mask is applied. This mask is typically configured to avoid masking BPDU packets and other protocol-conveying multicast packets.

Finally, a global destination mask is applied to unconditionally eliminate desired destinations.

Functional Description (continued)

Flow Identification

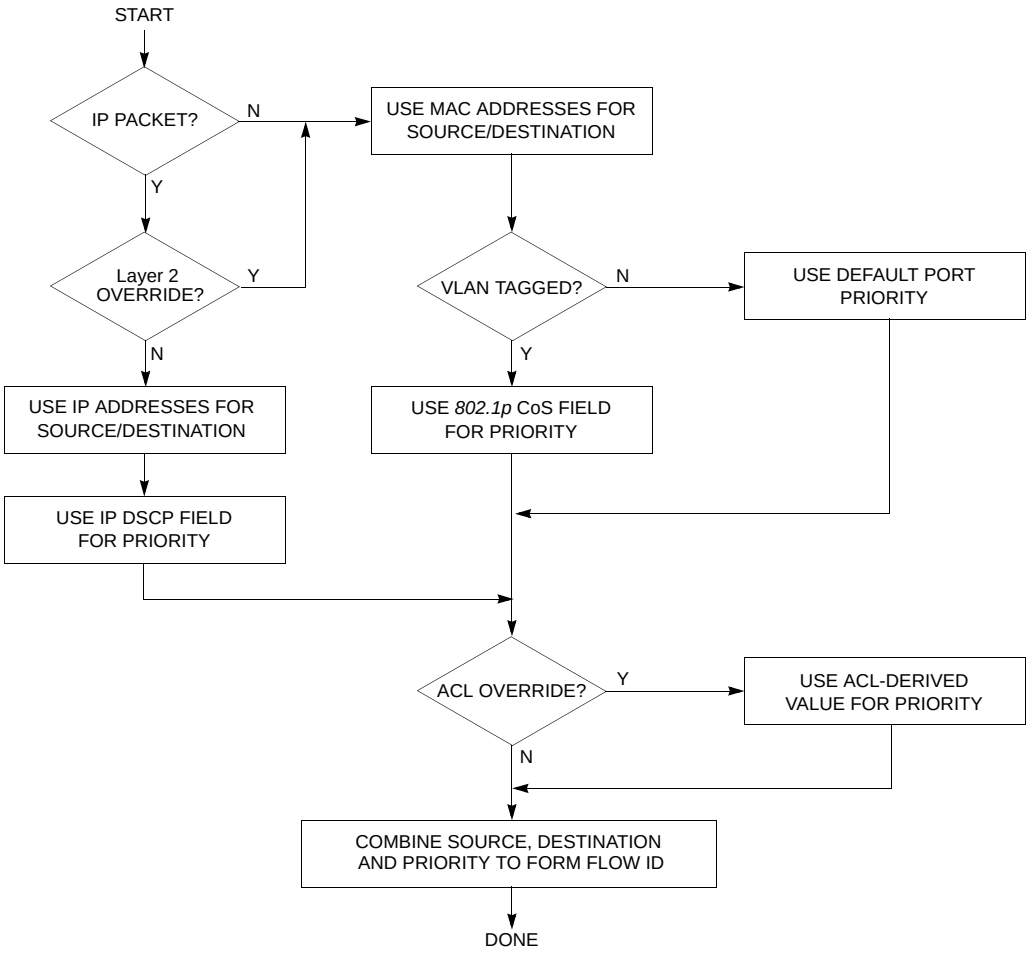


Figure 10. Flow Identification Process

All receive packets are assigned to one flow or another. Packets received via one of the Ethernet ports are assigned to one of eight flows per port. Each flow is associated with a policer that may be configured to limit the total bandwidth of the flows.

By default, IP packets utilize Layer 3 information to identify the flow to which the packet is assigned while non-IP packets utilize Layer 2 information. However, each port may be placed in a mode where Layer 2 information is utilized for flow classification for all packets, including IP packets.

If Layer 2 information is being utilized and the receive packet is VLAN tagged, then the CoS field from the VLAN tag is used to set the packet's priority. Otherwise, the receive port's default priority is used.

The result of the ACL processing for the receive packet has the ability to override any priority information that may be embedded in the packet or implied by its receive port.

Finally, the combination of the source, destination, and priority information is used to identify a single flow.

Functional Description (continued)

Policing

The ET3028-50 includes a series of policer functions that serve to limit the amount of ingress bandwidth permitted for any particular packet flow.

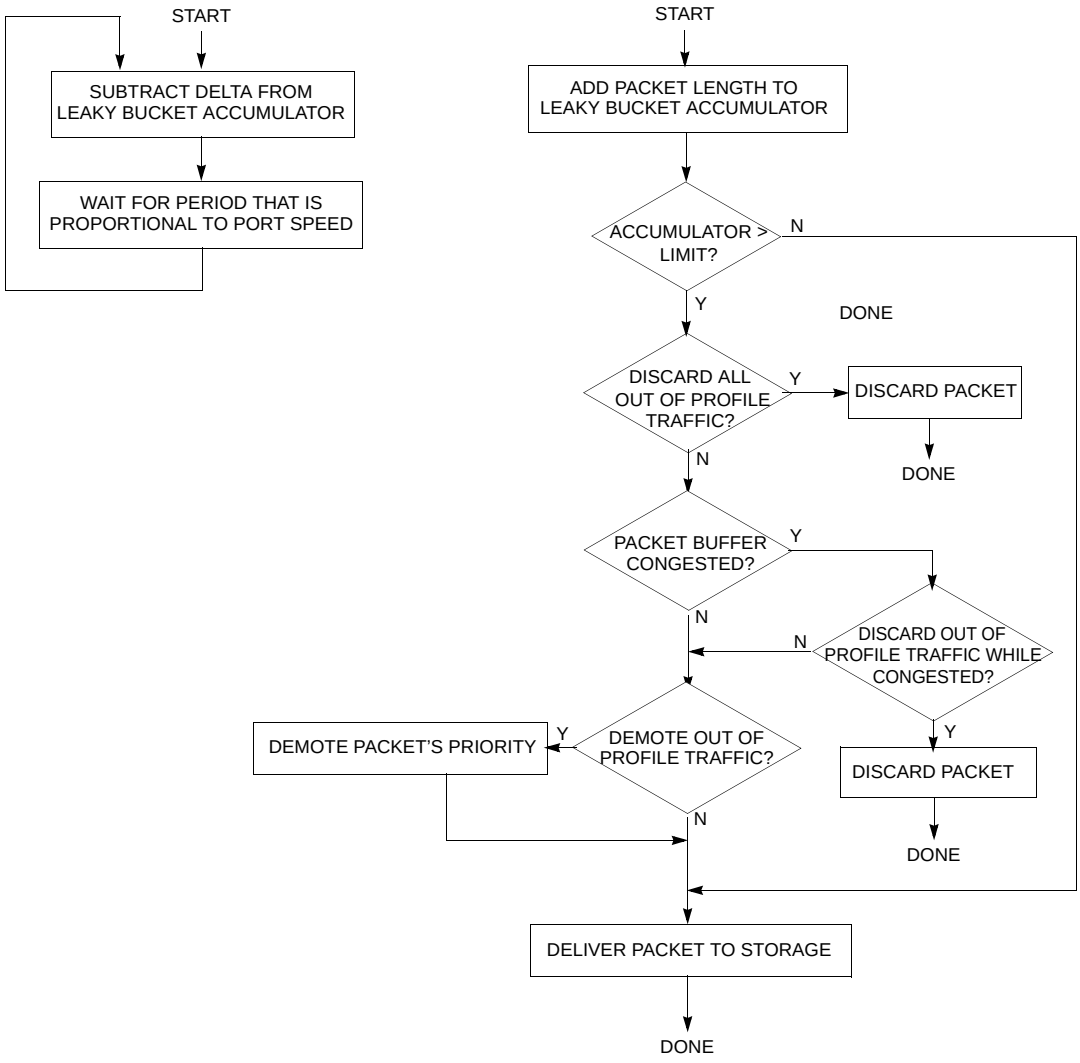


Figure 11. Policing Processes

Two processes run in parallel for each policer. The first process simply performs the leaky bucket function for the accumulator. The second process starts with each receive packet and determines if the receive packet is out of profile and, if so, which action to take.

The leaky bucket accumulator uses two parameters to regulate the leak rate. The first parameter is a delay interval. This delay interval is proportional to the operating speed of the associated receive port. i.e., the interval for a 10 Mbits/s port is 100 times that of a 1 Gbit/s port. This automatic proportional behavior means that the delta that is periodically applied to the accumulator can be reasonably thought of as a percentage of available bandwidth. The delta is adjusted as desired during configuration to establish the maximum receive bandwidth allowed for the associated traffic flow.

Functional Description (continued)

Policing (continued)

The accumulation process is triggered with each receive packet. The length (byte count) of each receive packet within the associated flow is added to the policer's accumulator. If the aggregate byte rate is greater than the bandwidth permitted by the leak delta, then the value in the accumulator will tend to increase. The traffic flow is not considered out of profile until the value in the accumulator exceeds the user-defined accumulator limit. The setting of this limit establishes the policer's burst tolerance. In other words, a higher limit means that the policer will tolerate a longer out-of-profile burst before declaring the flow out-of-profile.

Once the flow is declared out-of-profile, several options are available for dealing with individual packets within the flow. The ET3028-50 may be configured to discard all out-of-profile traffic. In this case, every packet in the flow is discarded for as long as the accumulator is greater than the limit.

Alternately, the ET3028-50 may be configured to only discard out-of-profile traffic if the packet buffer is indicating that it is congested. This congestion may be either global in nature (the total number of allocated buffers is greater than the global congestion threshold) or local (the number of buffers allocated to the associated receive port is greater that its congestion threshold).

Finally, the ET3028-50 has the ability to change the priority level of the receive packets for those flows that are out of profile. The priority demotion takes place during egress processing immediately prior to transmission.

Functional Description (continued)

Storage

During the storage process, packet buffer storage space is allocated to a receive packet. References to the packet are deposited onto the queues identified by the bridging and ACL processes.

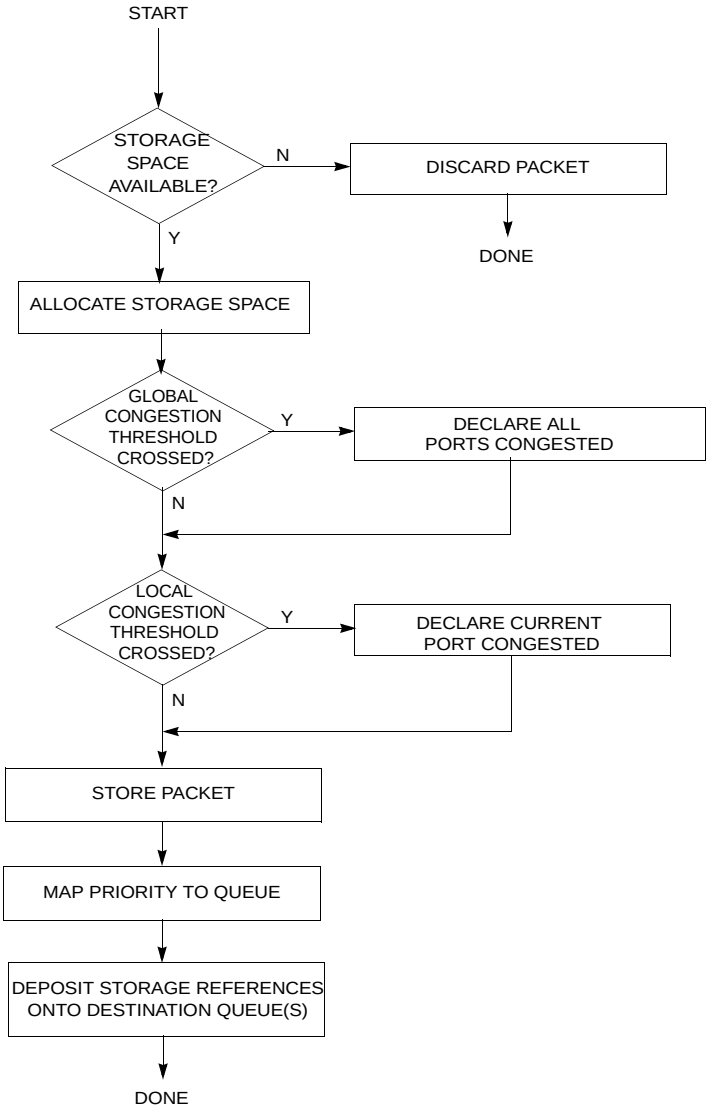


Figure 12. Storage Process

The first step in storing a receive packet is to ensure that sufficient storage resources are available to accommodate a maximum length packet. If insufficient resources are available, the packet is discarded.

If sufficient storage space is available, the space is allocated to the packet. This allocation may result in the packet buffer entering into a congested state, either globally or locally. If either condition is true, the appropriate indication is provided to the ingress policers and/or Ethernet MACs.

The remainder of the storage process entails the actual storage of the packet into the allocated buffers, mapping the packet's priority level (16 levels) to one of the eight queues that exist per transmit port. Finally, the queue mapping and the destination map are used to identify the various individual queues into which a reference to the stored packet is deposited.

Functional Description (continued)

Retrieval

Two processes per queue operate in parallel during packet retrieval. One process manages a bandwidth shaper function while the other performs arbitration and actually retrieves the stored packet.

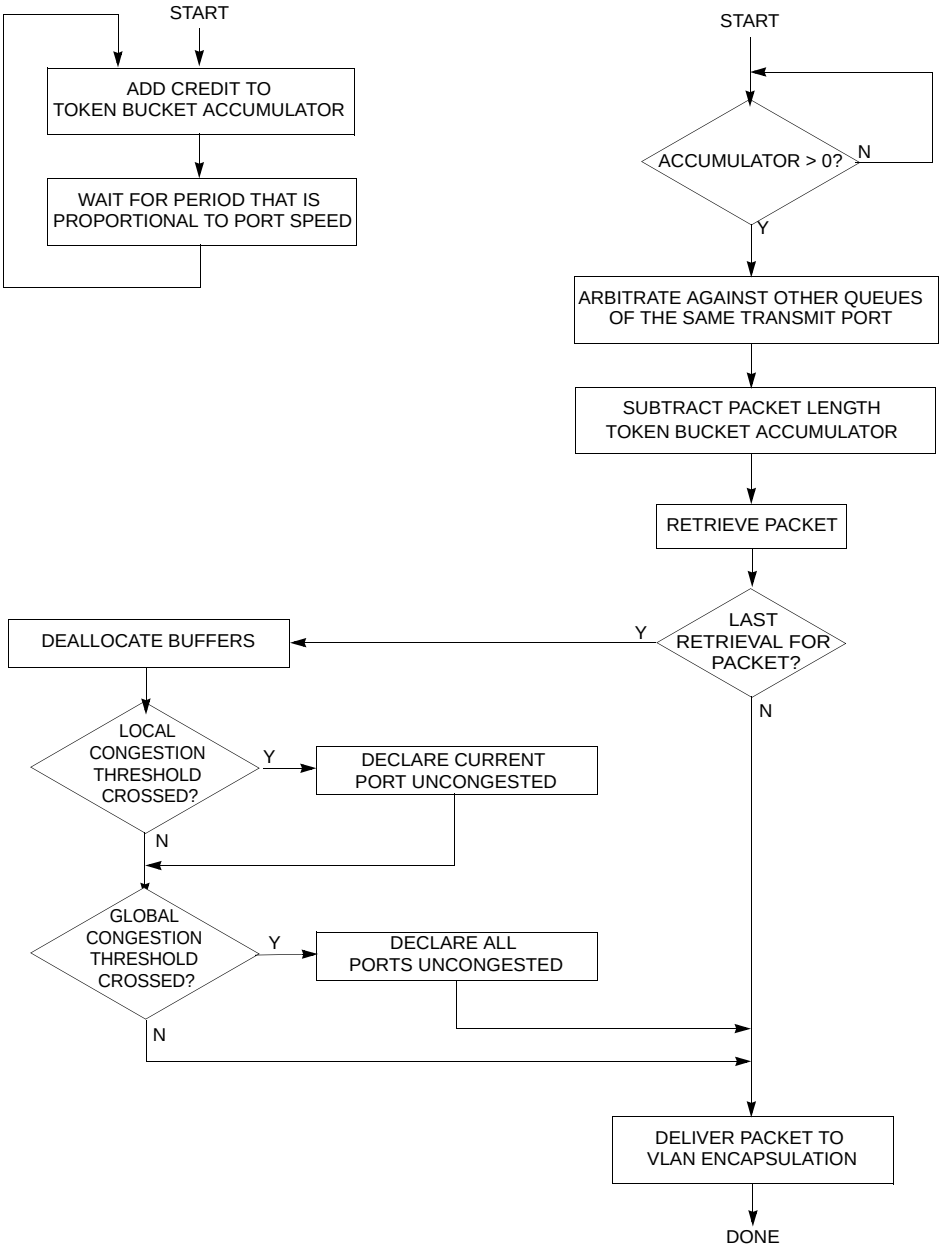


Figure 13. Retrieval Processes

Functional Description (continued)

Retrieval (continued)

A token bucket method is used for transmit bandwidth management. On a periodic basis, credit is applied to an accumulator. The period is set to be proportional to the port speed. In other words, the period of accumulation for a 10 Mbits/s Ethernet port is 100 times greater than the period for 1 Gbit/s Ethernet port. Hence, the credit applied each period is equivalent to a percentage of the total bandwidth available on a port. An accumulator limit is established that is used to set a maximum amount of credit that may be accumulated.

Whenever the accumulator is greater than zero, the associated queue is enabled to arbitrate against the other queues of the same transmit port. Each successful arbitration causes the length (byte count) of the transmit packet to be subtracted from the accumulator. The accumulator limit establishes the longest burst of transmit packet bytes that is allowed.

Packets may be destined for multiple queues or ports. This being the case, not every packet retrieval results in the deallocation of the buffer used for packet storage. Only after the completion of the last retrieval of a packet may its buffers be deallocated and made available for future storage operations.

The deallocation of buffers may result in the crossing of a congestion threshold towards the uncongested side. If this is the case, the appropriate congestion indications are removed.

Functional Description (continued)

VLAN Encapsulation

Prior to transmission, the VLAN encapsulation of the packet is updated as necessary. VLAN tags may be added, removed, or modified. Similarly, if a packet had its priority demoted by an ingress policer, its priority values (CoS and DSCP) may need to be replaced prior to transmission.

An optional mode is available whereby the VLAN encapsulation of the packet is not updated. If the packet is not updated, the packet will be transmitted with the same layer-2 header. If the packet was tagged, it will be transmitted with the same VLAN tag, and the COS will not be updated. If the packet was untagged, it will be transmitted without a VLAN tag. In both cases, the IP DSCP will not be updated.

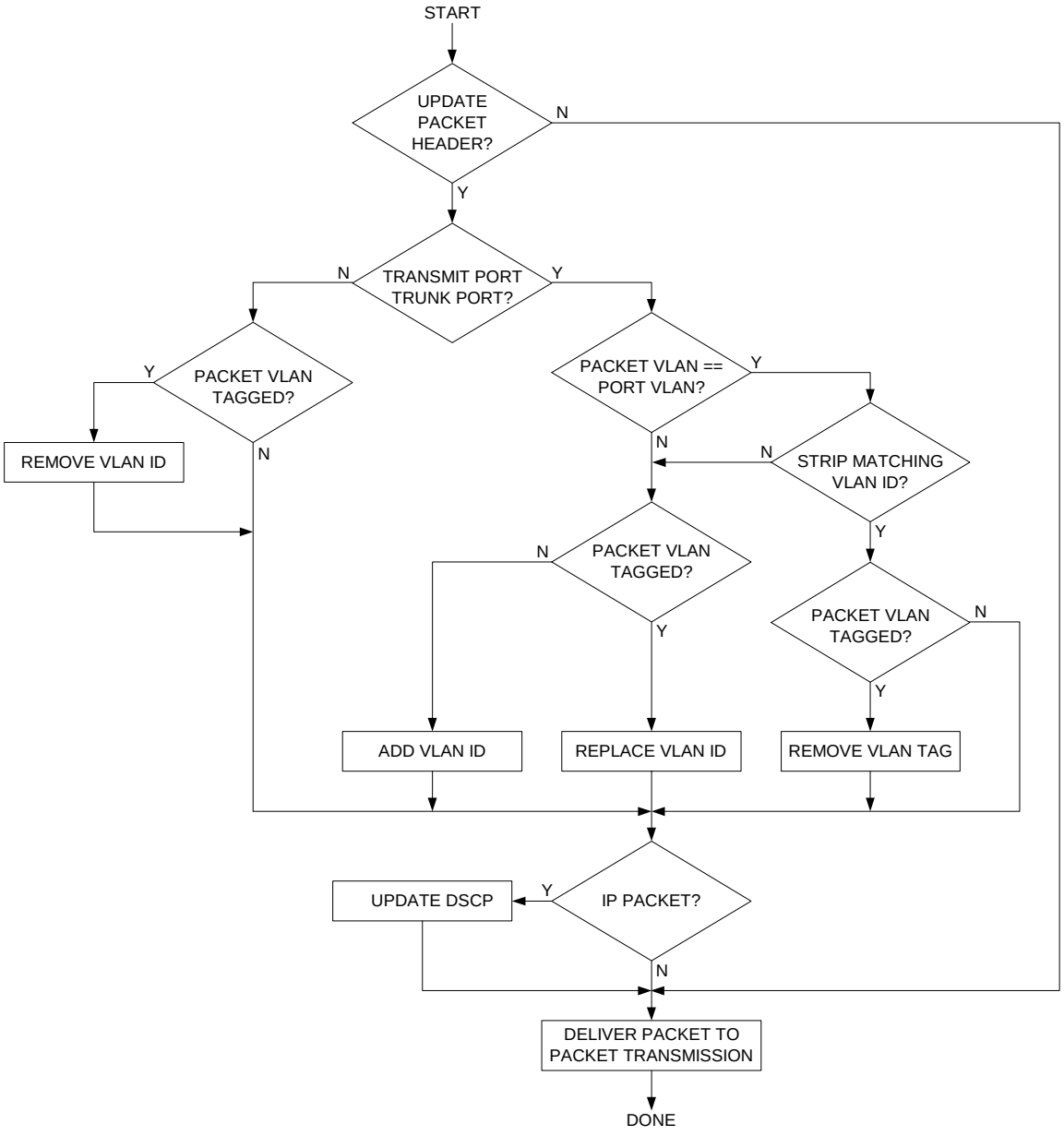


Figure 14. VLAN Encapsulation Process

Functional Description (continued)

VLAN Encapsulation (continued)

The VLAN encapsulation process operates in one of two fundamentally different ways depending on whether the transmit port is configured as a trunk port or an access port.

Access ports may only transmit packets that have no VLAN tags. Hence, if the packet was received with a VLAN tag, it must be removed from the packet prior to transmission.

The behavior of a trunk port is a bit more complex. An optional mode is available for the ET3028-50 whereby packets are transmitted on a trunk port without a VLAN tag if the VLAN that the packet belongs to matches the default VLAN ID of the transmit (trunk) port. If this mode is not in use, all packets transmitted by a trunk port are transmitted with a VLAN tag.

If the packet was received without a VLAN tag, then one must be added. If the packet was received with a VLAN tag, then its current tag is replaced with a new one. For typical configurations, the replacement VLAN tag is the same as the one that the packet was received with, and the replacement of the tag has no effect on the contents of the packet.

Finally, if the transmit packet is an IP packet, its DSCP value is updated prior to transmission. If the packet was marked for priority demotion during ingress policing, the new priority value reflects a reduction in priority. The nature of this reduction in priority depends on the configuration of the priority decode table.

Functional Description (continued)

Packet Transmission

The packet transmission process merges the regular network transmit packets with the full-duplex flow control packets and delivers the packets to the attached network.

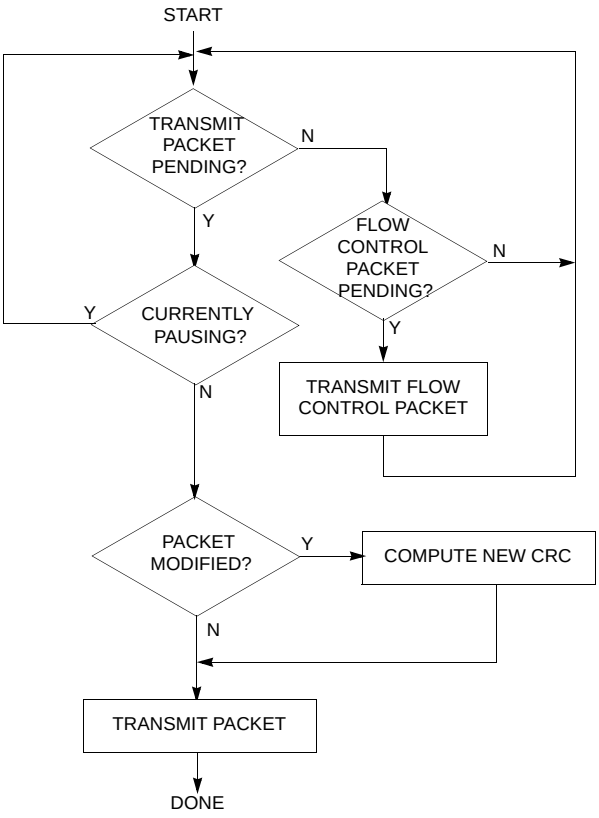


Figure 15. Packet Transmission Processes

Full-duplex flow control packets are given equal priority to normal transmit packets and are simply merged into the transmit flow on a round-robin basis. Flow control packets are only sent when the packet buffer is congested, and even then, they are not sent very often relative to the rate that normal packets may be transmitted.

If a packet was modified during its traversal of the ET3028-50, a new CRC value is computed by the transmitting Ethernet MAC. Otherwise, the CRC received with the packet is used for transmission.

Functional Description (continued)

Supervisor Packet Reception

Eight of the ET3028-50's 232 internal queues are dedicated to supervisor use. Normal bridging processes result in certain packets being forwarded to these queues. Typically, each queue is dedicated to a single purpose (e.g., source address learning) or a limited range of purposes.

The Layer 2 address tables may be configured to forward or copy any arbitrary MAC destination address to the supervisor. Also, certain types of packets are generally forwarded or copied to the supervisor. These packet types include those with unknown MAC source addresses, IGMP packets, ACL logged packets, and others. Generally, each of the supervisor's eight queues are dedicated to a very limited number of traffic types.

The packet queuing and buffering space within the ET3028-50 is limited, and the supervisor's ability to keep up with certain types of traffic may be very unlikely. Hence, the ET3028-50 transfers packets from its internal buffers to buffers created within the supervisor's memory system across the ET3028-50's *PCI* bus. The ET3028-50 does this as a *PCI* initiator.

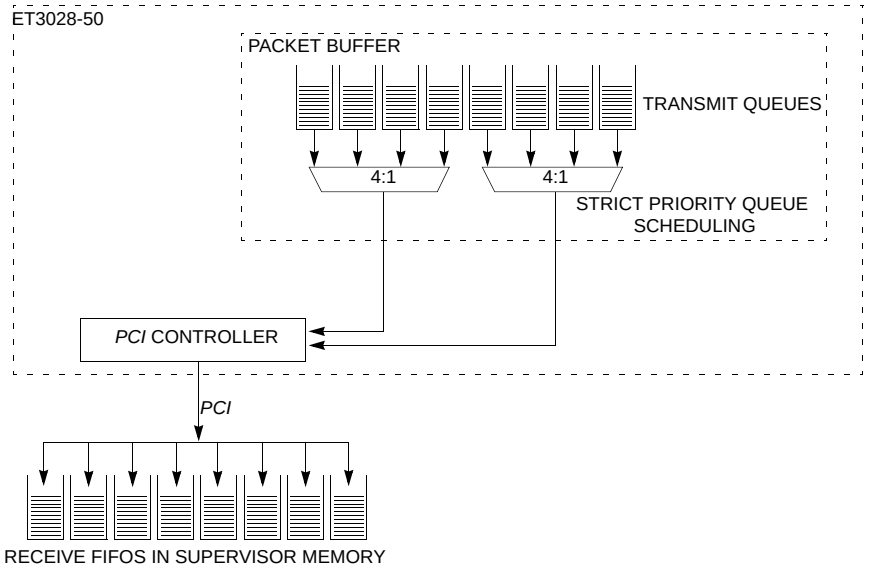


Figure 16. Supervisor Packet Reception Structures

Figure 16 shows how eight transmit queues are dedicated to the supervisor. These eight particular transmit queues (destination map ports 50 through 57) are associated in a one-to-one relationship with eight receive FIFOs that are established in the supervisor's memory space.

Functional Description (continued)

Supervisor Packet Reception (continued)

Data Structures

The receive structures established within the supervisor's memory system consist of a series of circular FIFOs. The starting and ending locations of the FIFOs are defined through a series of registers within the ET3028-50 (`Supervisor_Rx_Fifo_Limits_{0..7}`). These starting and ending locations of each FIFO remain static once configured by the supervisor.

Read and write pointers (`Supervisor_Rx_Fifo_Ptr_{0..7}`) are used during the operation of the FIFOs. The ET3028-50 advances the write pointers as packet data is stored in the FIFO space, and the supervisor advances the read pointer as it processes and discards the packets.

All packets transferred into the supervisor's receive queues are treated integrally. In other words, if the space remaining in the FIFO before its physical endpoint (not to be confused with its logical endpoint) is not sufficient to accommodate a maximum length packet, then the packet is stored at the beginning of the FIFO's physical range.

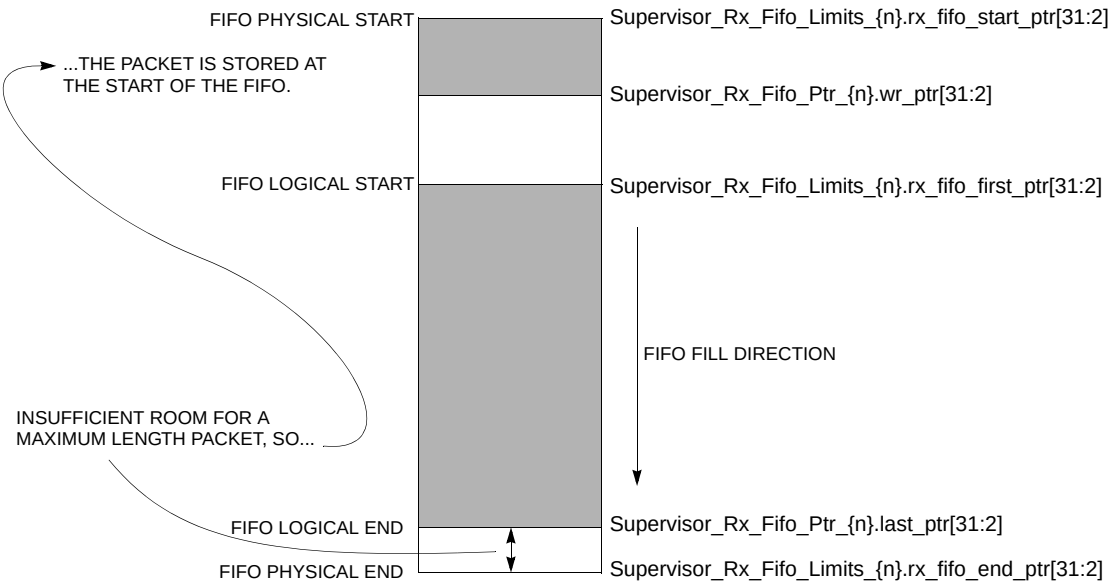


Figure 17. Receive Circular FIFO Structure

The discontinuity caused by the gap left at the end of the physical range of the FIFO is handled by means of a 32-bit pointer that is written to the FIFO as the first word of the packet's data structure. The 32-bit pointer (`Supervisor_Rx_Packet.packet_start_ptr[31:2]`) points to where the packet can actually be found. Normally, this pointer points to the very next 32-bit word (as in the case of contiguously spaced receive packets). However, when a discontinuity is introduced as a consequence of keeping packets integral, the pointer points to the start of the FIFO's physical range rather than the next 32-bit word location.

Functional Description (continued)

Supervisor Packet Reception (continued)

Packet Reception Process

Packets received by the physical Ethernet interfaces may be forwarded or copied to one or more of the queues dedicated to the supervisor. These forwarding and copying decisions are based on the contents of the packets.

These supervisor queues are serviced in a normal manner. However, rather than being directed to Ethernet ports, these packets are directed to the supervisor via the *PCI* bus.

The identity of each packet's source queue accompanies the packet data as it is retrieved and forwarded to the supervisor. This identifying information is used to determine which of the eight receive FIFOs in supervisor memory are to receive the packets.

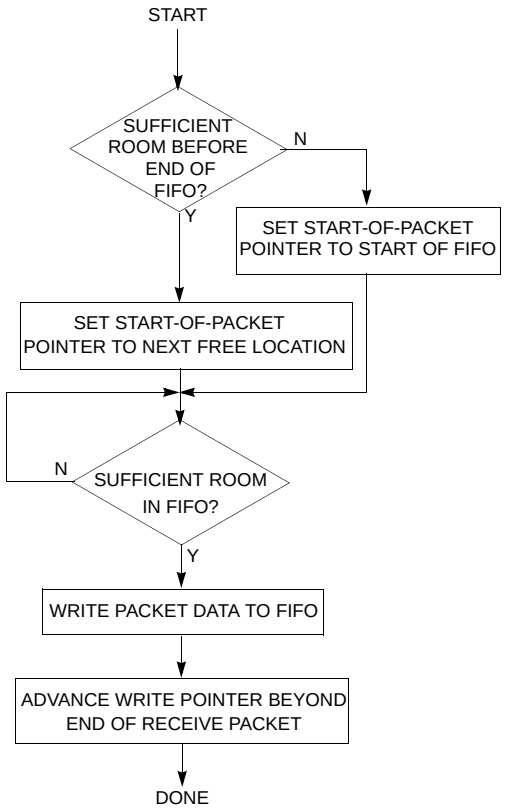


Figure 18. Supervisor Packet Reception Process (Hardware Actions)

Functional Description (continued)

Supervisor Packet Reception (continued)

Hardware Actions.

1. Check if close to the end of the FIFO's physical limit.

The first step taken by the ET3028-50 is to determine if the space remaining between the logical end of a receive FIFO and its physical end (refer to Figure 16 on page 35 for a depiction of the terms) is great enough to accommodate a maximum length packet. The following equation is used to make this determination:

```
( Supervisor_Rx_Fifo_Limits_{0..7}.rx_fifo_end_ptr[31:0] -
Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2] ) >=
Supervisor_Mode.maximum_packet_length[13:0]
```

If the preceding equation is true, then proceed to step 2. Otherwise, proceed to step 3.

2. Set start-of-packet pointer in the packet data structure to the subsequent location.

Here, `Supervisor_Rx_Packet.packet_start_ptr[31:2]` is set equal to `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2] + 4` (i.e., the next memory location). `Supervisor_Rx_Packet.packet_start_ptr[31:2]` is then written to the supervisor memory location pointed to by `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]` and the write pointer is advanced[1] by 4.

Go to step 4.

3. Set start-of-packet pointer in the packet data structure to the start of the FIFO.

If the preceding equation is false, then `Supervisor_Rx_Packet.packet_start_ptr[31:2]` is set equal to `Supervisor_Rx_Fifo_Limits_{0..7}.rx_fifo_start_ptr[31:2]`. `Supervisor_Rx_Packet.packet_start_ptr[31:2]` is then written to the supervisor memory location pointed to by `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]` and the write pointer is set equal to `Supervisor_Rx_Fifo_Limits_{0..7}.rx_fifo_start_ptr[31:0]`.

Continue with step 4.

4. Check FIFO capacity.

The last check is to determine if there is sufficient room between `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]` and `Supervisor_Rx_Fifo_Limits_{0..7}.rx_fifo_first_ptr[31:2]` to store a maximum length packet. The following equation is used to make this determination[2]:

```
( Supervisor_Rx_Fifo_Limits_{0..7}.rx_fifo_first_ptr[31:2] -
Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2] ) >=
Supervisor_Maximum_Packet_Length.maximum_packet_length[13:0]
```

If the preceding equation is false, wait. Otherwise, proceed to step 5.

5. Write packet data.

If the preceding equation is true, then there is sufficient room for a maximum length packet and the packet transfer commences. `Supervisor_Rx_Packet[4..<end of packet>]` is written to sequential 32-bit words of supervisor memory with `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]` advancing after each write. At the completion of the transfer, `Supervisor_Rx_Fifo_Ptr_{0..7}.last_ptr[31:2]` is set equal to the value that `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]` held prior to any writes. This last step marks the first 32-bit word of the last packet in the receive FIFO.

Whenever `Supervisor_Rx_Fifo_Limts_{0..7}.first_ptr[31:2]` and `Supervisor_Rx_Fifo_Ptr_{0..7}.last_ptr[31:2]` are not equal, then the FIFO is not empty and an indication to that effect is provided to the supervisor via `Supervisor_Rx_Fifo_Status_{0..7}.not_empty`.

1. The write pointer (`Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]`) may never be advanced by the hardware to be equal to the first pointer (`Supervisor_Rx_Fifo_Limits_{0..7}.rx_fifo_first_ptr[31:2]`). If advancing the first pointer would cause such a situation, the advancement of the write pointer is delayed until the software has moved the first pointer.
2. If the difference computed in this equation is negative, then the equation is considered to be true.

Functional Description (continued)

Supervisor Packet Reception (continued)

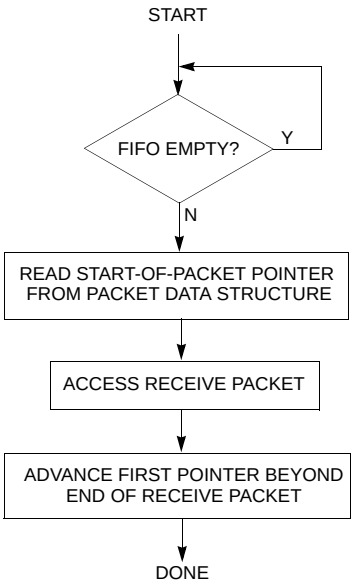


Figure 19. Supervisor Packet Reception Process (Supervisor Actions)

Supervisor Actions. Upon detecting that `Supervisor_Rx_Fifo_Status_{0..7}.not_empty is true`, the supervisor reads `Supervisor_Rx_Fifo_Ptr_{0..7}.first_ptr[31:2]` in order to determine the location of the start of the first packet in the receive FIFO.

Reading the memory location pointed to by `Supervisor_Rx_Fifo_Ptr_{0..7}.first_ptr[31:2]` returns `Supervisor_Rx_Packet.packet_start_ptr[31:2]`, the pointer to the remainder of the packet. The supervisor then begins reading the remainder of the packet starting at the indicated location.

When the supervisor has finished working with the current packet, it writes the address of the first 32-bit word beyond the end of the packet to `Supervisor_Rx_Fifo_Ptr_{0..7}.first_ptr[31:2]`. This has the effect of allowing the ET3028-50 to overwrite the packet.

Functional Description (continued)

Supervisor Packet Transmission

The ET3028-50 supervisor processor performs packet transmissions by faking packet receptions.

The supervisor presents a packet to the ET3028-50's receive packet processing as if it were a normal receive packet. However, rather than using look-up results derived from the packet's contents, the Layer 2 and ACL look-up operations use look-up results that have been queued by the supervisor in advance of reception of the packet. These supervisor-supplied look-up results are used to direct the packet to the queues associated with the transmit ports desired by the supervisor.

Because of this requirement to preload the look-up results, the transmit packet data structure is headed by a series of fields that hold such information as a transmit port map, priority level, and the like.

Two independent queues are available to the supervisor for packet transmission. The two queues have a strict priority relationship. Whenever the high-priority queue is not empty, it preempts the operations of the low-priority queue.

Data Structures

The following three sets of information stored in the supervisor's memory are necessary for packet transmission:

1. Packet transmission FIFO.
2. Packet descriptor blocks.
3. Packet segments.

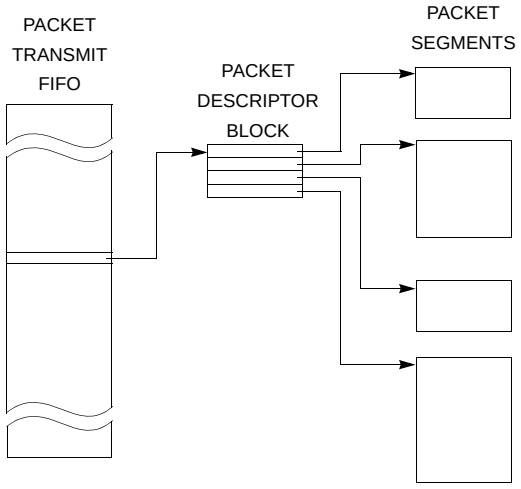


Figure 20. Supervisor Packet Transmission Data Structures

Functional Description (continued)

Supervisor Packet Transmission (continued)

Packet Transmission Queue

Two packet transmission queues are maintained within supervisor memory. Each queue consists of a series of 32-bit words that are pointers to packet descriptor blocks. One queue entry and, consequently, one packet descriptor block, corresponds to a single packet. A single packet descriptor block may have more than one reference in the packet transmission queues.

The physical limits of the packet transmission queues as well as their read and write pointers are defined by a series of registers.

Packet Descriptor Blocks. Packet descriptor blocks are variable length lists of pointers and byte counts that identify the packet segments to be gathered up to form a transmit packet. By allowing packets to be segmented into several pieces, the supervisor has the option of having catalogs of look-up result headers, Layer 2 headers, and Layer 3 headers that can be concatenated to form a packet's header. Packet bodies may also be cataloged and warehoused in this fashion.

Each packet descriptor block consists of a one or more pairs of 32-bit words. The first 32-bit word is a pointer to a packet segment. The second 32-bit word contains a byte count for the segment and a flag that indicates whether or not the current segment is the last segment for the packet. The processing of these descriptor block records continues until a last flag is encountered.

Packet Segments. Packet segments are simple byte strings that make up the actual packet data. Packet segments may range in size from a single byte to 16 Kbytes.

Functional Description (continued)

Supervisor Packet Transmission (continued)

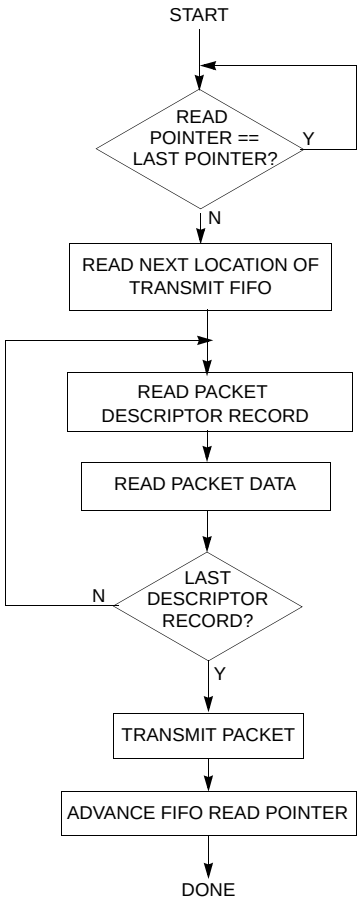


Figure 21. Supervisor Packet Transmission Process (Hardware Actions)

Packet Transmission Process

The supervisor must examine the state of the transmit FIFO in order to determine what to set `Supervisor_Tx_Fifo_Ptr_{0..1}.last[31:2]` to. If the current location pointed to by `Supervisor_Tx_Fifo_Ptr_{0..1}.last[31:2]` is one location behind `Supervisor_Tx_Fifo_Ptr_{0..1}.rd_ptr[31:2]`, then the FIFO is full and the supervisor must wait until space has been made available.

If `Supervisor_Tx_Fifo_Ptr_{0..1}.last[31:2]` is equal to `Supervisor_Tx_Fifo_Limits_{0..1}.end[31:2]`, then the supervisor must make the location pointed to by `Supervisor_Tx_Fifo_Limits_{0..1}.start[31:2]` as the next FIFO location utilized.

The supervisor then sets up the packet's data structures as shown in Figure 20 on page 40. It then sets `Supervisor_Tx_Fifo_Ptr_{0..1}.last[31:2]` to point to the new entry in `Supervisor_Tx_Fifo`.

The ET3028-50 detects that `Supervisor_Tx_Fifo_Ptr_{0..1}.rd_ptr[31:2]` and `Supervisor_Tx_Fifo_Ptr_{0..1}.last[31:2]` are no longer equal and advances `Supervisor_Tx_Fifo_Ptr_{0..1}.rd_ptr[31:2]` to the next location within `Supervisor_Tx_Fifo`. This location is then read by the ET3028-50 to gather the pointer to the transmit packet's descriptor block.

Functional Description (continued)

Supervisor Packet Transmission (continued)

Packet Transmission Process (continued)

Pairs of 32-bit words are read from `Supervisor_Tx_Descriptor` in order to determine the location and byte count of each of the packet's segments. If `Supervisor_Tx_Descriptor.last` is asserted, then the current segment is the last segment of the packet.

The ET3028-50 reads the packet segments in the order in that their descriptors appear in `Supervisor_Tx_Descriptor`. The first segment of every transmit packet contains the destination port map and priority information for the packet.

The packet segments are stored as a contiguous packet within the ET3028-50 in the normal manner and queued for transmission utilizing the supplied destination information.

Data Structures

Supervisor_Rx_Fifo_{0..7}

Description: An assortment of receive packets.

Table 7. Supervisor_Rx_Fifo_{0..7} Register Parameters

Parameter	Value
Base Address	`Supervisor_Rx_Fifo_Limits_{0..1}.start[31:2]`
Structure Size	`Supervisor_Rx_Fifo_Limits_{0..1}.end[31:2] – Supervisor_Rx_Fifo_Limits_{0..1}.start[31:2] + 4`
Structure Instances	8
Structure Spacing	Variable

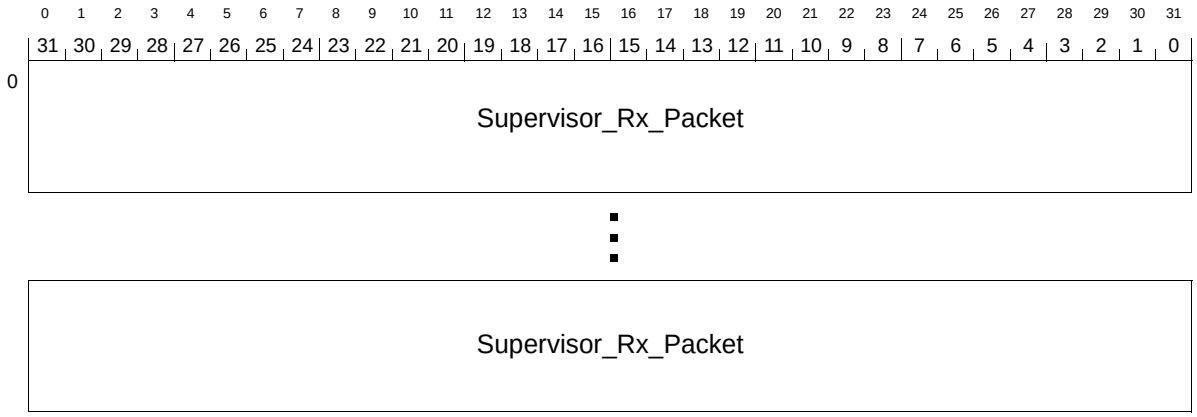


Figure 22. Supervisor_Rx_Fifo_{0..7} Data Structure

This data structure is a collection of `Supervisor_Rx_Packet` data structures.

The physical extent of `Supervisor_Rx_Fifo{0..7}` is defined by `Supervisor_Rx_Fifo_Limits_{0..7}`.

Data Structures (continued)

Supervisor_Rx_Packet

Description: The form of received packets in supervisor memory space.

Table 8. Supervisor_Rx_Packet Register Parameters

Parameter	Value
Base Address	One 32-bit word after the previous `Supervisor_Rx_Packet`
Structure Size	Variable
Structure Instances	Variable
Structure Spacing	Variable

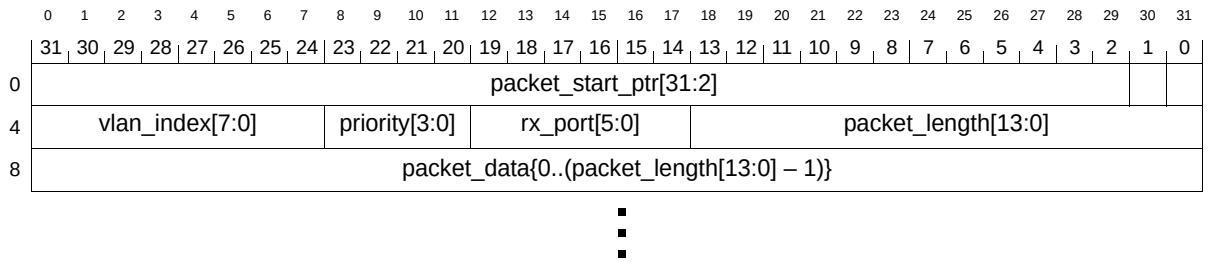


Figure 23. Supervisor_Rx_Packet Data Structure

Data Structures (continued)

Supervisor_Rx_Packet (continued)

Table 9. Supervisor_Rx_Packet Field Parameters

Field Name	Parameter	Description
packet_start_ptr[31:2]	Mode = RO Offset = 0.0	Points to the status word of the associated receive packet. Ordinarily, `packet_start_pointer[31:2]` and `{rx_port[5:0], packet_length[13:0]}` are adjacent. However, in order to maintain integral packets, these two words may be nonadjacent near the end of the physical extent of a supervisor's receive FIFO. This value is written by the ET3028-50 and read by the supervisor.
truncated	Mode = RO Offset = 0.30	If a receive packet is truncated because its length exceeds that set in `Supervisor_Maximum_Packet_Length`, then this bit is asserted.
parity_error	Mode = RO Offset = 4.0	This bit is asserted if an internal parity error is detected at any time during the transfer of a packet from the ET3028-50 across the *PCI* bus to the supervisor.
vlan_index[7:0]	Mode = RO Offset = 4.0	The VLAN index assigned to a packet during reception.
priority[3:0]	Mode = RO Offset = 4.8	The priority value assigned to a packet during reception.
rx_port[5:0]	Mode = RO Offset = 4.12	This value identifies the physical port via which a packet was received. This value is written by the ET3028-50 and read by the supervisor.
packet_length[13:0]	Mode = RO Offset = 4.18	The length of a received packet. `packet_length[13:0]` is limited by `Supervisor_Maximum_Packet_Length`. If `packet_length[13:0]` equals `Supervisor_Maximum_Packet_Length`, then the packet may have been truncated during the transfer. This value is written by the ET3028-50 and read by the supervisor.
packet_data	Mode = RO Offset = 8.0	The packet data. Packet data is arranged in a big-endian fashion. The last 32-bit word may contain zero or more pad bytes. This value is written by the ET3028-50 and read by the supervisor.

This data structure is used to present received packets to the supervisor. Zero or more of these data structures are arranged within a FIFO structure maintained within the supervisor's memory. The first word of `Supervisor_Rx_Packet` always immediately follows the last word of the previous instance.

The first word of this data structure is a pointer to the remainder of the data structure. Ordinarily, these structures are arranged contiguously. If the space at the end of `Supervisor_Rx_Fifo{0..7}` is insufficient to accommodate a maximum length packet, then `Supervisor_Rx_Packet.packet_start_ptr[31:2]` is used to point to the start of `Supervisor_Rx_Fifo{0..7}`.

If there is insufficient space to accommodate a maximum length packet anywhere in `Supervisor_Rx_Fifo{0..7}`, then the FIFO is considered full even though a smaller packet may fit.

If `Supervisor_Rx_Fifo{0..7}` is empty, then `Supervisor_Rx_Packet` is placed at `Supervisor_Rx_Fifo_Ptr_{0..7}.wr_ptr[31:2]`.

Data Structures (continued)

Supervisor_Tx_Descriptor

Description: A list of descriptors of supervisor transmit packet segments.

Table 10. Supervisor_Tx_Descriptor Register Parameters

Parameter	Value
Base Address	`Supervisor_Tx_Fifo_{0..1}.tx_descriptor_ptr[31:2]`
Structure Size	Variable
Structure Instances	Variable
Structure Spacing	Variable

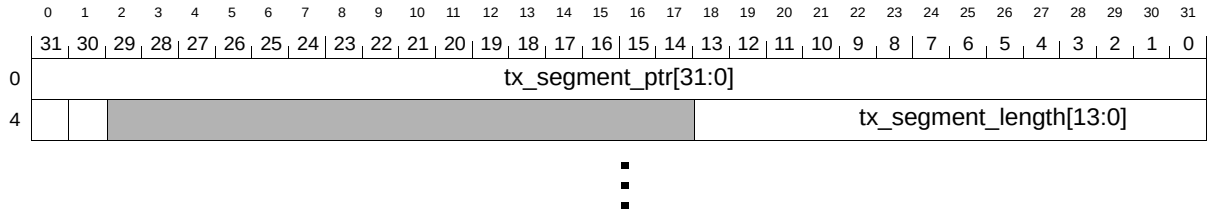


Figure 24. Supervisor_Tx_Descriptor Data Structure

Table 11. Supervisor_Tx_Descriptor Field Parameters

Field Name	Parameter	Description
tx_segment_ptr[31:0]	Mode = R/W Offset = 0.0	A pointer to a supervisor transmit packet segment.
last	Mode = R/W Offset = 4.0	When asserted, this descriptor is a packet's last descriptor.
indicate	Mode = R/W Offset = 4.1	When asserted by the supervisor, the `tx_packet` indication is asserted in `Supervisor_Ind` upon completion of the transfer of the corresponding transmit packet segment.
tx_segment_length[13:0]	Mode = R/W Offset = 4.18	The size of a packet segment.

This data structure is a list of descriptors of packet segments. Each packet segment requires its own descriptor. The descriptors are processed in the order that they are encountered in `Supervisor_Tx_Descriptor`. Each packet requires a `Supervisor_Tx_Descriptor` structure with at least one record.

Transmit packet processing is terminated upon the completion of processing of the record whose `last` bit is asserted.

`tx_segment_ptr[31:0]` is able to address segments on any arbitrary byte boundary. Each segment may also be of any arbitrary byte length. The ET3028-50 ensures that packet segments are stitched together correctly.

The physical extent of `Supervisor_Tx_Descriptor` is defined by `Supervisor_Tx_Fifo_Limits_{0..1}`.

Data Structures (continued)

Supervisor_Tx_Fifo_{0..1}

Descriptions: FIFO list of transmit descriptor block pointers.

Table 12. Supervisor_Tx_Fifo_{0..1} Register Parameters

Parameter	Value
Base Address	`Supervisor_Tx_Fifo_Limits_{0..1}.start_ptr[31:2]`
Structure Size	`Supervisor_Tx_Fifo_Limits_{0..1}.end_ptr[31:2] - Supervisor_Tx_Fifo_Limits_{0..1}.start_ptr[31:2] + 4`
Structure Instances	2
Structure Spacing	Variable

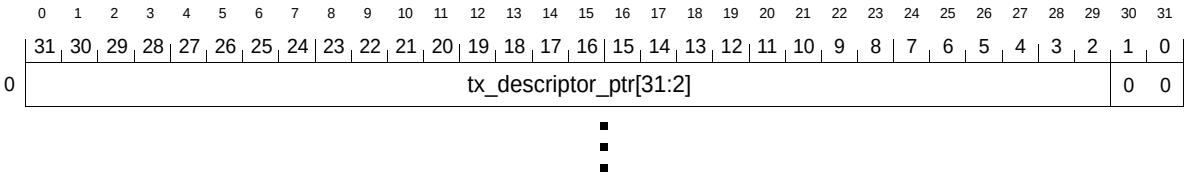


Figure 25. Supervisor_Tx_Fifo_{0..1} Data Structure

Table 13. Supervisor_Tx_Fifo_{0..1}

Field Name	Parameter	Description
tx_descriptor_ptr[31:2]	Mode = R/W Offset = 0.0	A pointer to a supervisor transmit packet descriptor block.

This data structure is a collection of pointers to transmit descriptor blocks.

The physical extent of `Supervisor_Tx_Fifo{0..1}` is defined by `Supervisor_Tx_Fifo_Limits_{0..1}`.

Data Structures (continued)

Supervisor_Tx_Packet

Description: A string of transmit packet data bytes.

Table 14. Supervisor_Tx_Packet Register Parameters

Parameter	Value
Base Address	Variable (multiple of 4)
Structure Size	Variable
Structure Instances	Variable
Structure Spacing	Variable

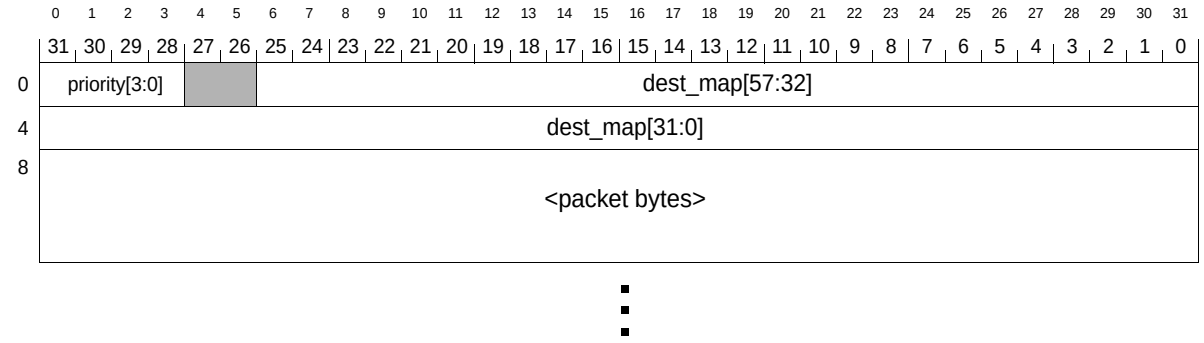


Figure 26. Supervisor_Tx_Packet Data Structure

Table 15. Supervisor_Tx_Packet

Field Name	Parameter	Description
priority[3:0]	Mode = R/W Offset = 0.0	The packet's priority value.
dest_map[57:0]	Mode = R/W Offset = 0.6	The packet's destination map. Asserted bits correspond to selected transmit ports.
<packet bytes>	Mode = R/W Offset = 8.0	The transmit packet data bytes.

This data structure is a single transmit packet. Every transmit packet is preceded by a `priority[3:0]` value and a `dest_map[57:0]` vector. The packet data bytes start at byte offset 8 and continue from there. The byte located at offset 8 must always be the first byte of packet's 48-bit Layer 2 destination address field.

The supervisor must not included a CRC value with the packet. A CRC is automatically calculated and appended during the transmission process.

Data Structures (continued)

Supervisor_Tx_Packet_Segment

Description: A segment of a transmit packet.

Table 16. Supervisor_Tx_Packet_Segment Register Parameters

Parameter	Value
Base Address	Supervisor_Tx_Descriptor.tx_segment_ptr[31:0]
Structure Size	Supervisor_Tx_Descriptor.tx_segment_length[13:0]
Structure Instances	Variable
Structure Spacing	Variable

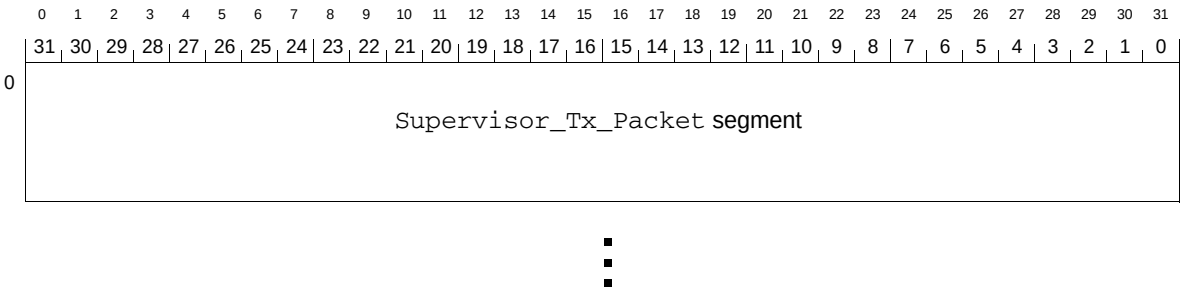


Figure 27. Supervisor_Tx_Packet_Segment Data Structure

This data structure is a single segment of a packet. One or more segments are used to assemble transmit packets.

Electrical Specifications

Absolute Maximum Ratings

Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operations sections of the data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

Table 17. Absolute Maximum Ratings

Parameter	Symbol	Max	Unit
Core Supply Voltage	VDD12_CORE VDD_4	1.7	V
Core PLL Voltage	AVDD12_PLL_CORE	1.7	V
SGMII Termination Voltage	VDD15L	4.6	V
SGMII I/O Voltage	VDD33L	4.6	V
SGMII PLL Voltage	AVDD12_PLL	1.7	V
SFP Output Buffer Voltage	VDDOB[27:24]	1.7	V
SFP Core Voltage	VDD_3	1.7	V
SFP Input Buffer Voltage	VDDIB[27:24]	1.7	V
PCI I/O Voltage	VDD33	4.6	V
Storage Temperature	T_{stg}	125	°C

ESD Protection

Table 18. ESD Protection

Parameter	Value	Unit
High-Speed Pins		
HBM	1.5	kV
CDM	250	V
Low-Speed Pins		
HBM	2.0	kV
CDM	500	V

Electrical Specifications (continued)

Recommended Operating Conditions

Table 19. Recommended Operating Conditions*

Parameter	Symbol	Min	Typ	Max	Unit
Core Supply Voltage	VDD12_CORE VDD_4	1.14	1.2	1.26	V
Core PLL Voltage	AVDD12_PLL_CORE	1.14	1.2	1.26	V
SGMII Termination Voltage	VDD15L	1.425	1.5	1.575	V
SGMII I/O Voltage	VDD33L	3.135	3.3	3.465	V
SGMII PLL Voltage	AVDD12_PLL	1.14	1.2	1.26	V
SFP Output Buffer Voltage	VDDOB[27:24]	1.425	1.5	1.575	V
SFP Core Voltage	VDD_3	1.14	1.2	1.26	V
SFP Input Buffer Voltage	VDDIB[27:44]	1.425	1.5	1.575	V
PCI I/O Voltage	VDD33	3.135	3.3	3.465	V
Ambient Operating Temperature	T_A	0	—	70	°C

* All voltages are ±5%.

Power Supply Consumption

Table 20. Power Supply Consumption

Power Supply	Maximum Current Consumption per Supply
1.2 V Supply	
VDD12_CORE	5180 mA
AVDD12_PLL	
AVDD12_PLL_CORE	
VDD_3	
VDD_4	
1.5 V Supply	
VDD15L	240 mA
VDDIB_(27:24)	
VDDOB_(27:24)	
3.3 V Supply	
VDD33L	330 mA
VDD33	

Thermal Characteristics

Table 21. Thermal Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
Junction Temperature	T_J	—	112	125	°C
Theta Junction to Case	θ_{JC}	—	0.46	—	°C/W
Theta Case to Ambient Required by Heat Sink	θ_{CA}	—	2.91	—	°C/W

Electrical Specifications (continued)

PCI I/O Specification

Table 22. dc Electrical Specification—*PCI*

Parameter	Symbol	Pin Type	Conditions	Min	Max	Unit
Input High Voltage	V_{IH}	I	—	0.5 x VDD33_IO	VDD33_IO + 0.5	V
Input Low Voltage	V_{IL}	I	—	–0.5	0.35 x VDD33_IO	V
Input High Current	I_{IH}	I	—	—	—	
Input Low Current	I_{IL}	I	—	—	—	
Output High Voltage	V_{OH}	O	I_{OH} = –0.5 mA	0.9 x VDD33_IO	—	V
Output Low Voltage	V_{OL}	O	I_{OL} = 1.5 mA	—	0.1 x VDD33_IO	V
Tristate Leakage	I_{OZ}	O	0 < V_{IN} < VDD33_IO	–10	10	µA
Input Leakage	I_{IZ}	I	0 < V_{IN} < VDD33_IO	–1	1	µA

JTAG I/O Specification

Table 23. dc Electrical Specification—JTAG

Parameter*	Symbol	Pin Type	Conditions	Min	Typ	Max	Unit
Maximum Input High Voltage	V_{IMAX}	I	—	—	—	5.5	V
Minimum Input Low Voltage	V_{ILMAX}	I	—	0.0	—	—	V
Input High Voltage	V_{IH}	I	—	1.3	1.55	2.0	V
Input Low Voltage	V_{IL}	I	—	0.8	1.20	1.5	V
Input Hysteresis	V_{HYS}	I	—	0.3	0.35	0.4	V
Output High Voltage	V_{OH}	O	—	2.4	—	—	V
Output Low Voltage	V_{OL}	O	—	0.4	—	—	V
Output High Current	I_{OH}	O	—	—	10	—	mA
Output Low Current	I_{OL}	O	—	—	10	—	mA
Tristate Leakage Current	—	O	V_{IN} = 0 V to 5.5 V	–10	±1	10	µA
Output Resistance	R_O	O	—	16	21	34	Ω

* All parameters measured at R_{LOAD} = 100 Ω ± 1%.

Electrical Specifications (continued)

SGMII I/O Transmit Specifications

Table 24. SGMII I/O Transmit Specifications

Parameter	Symbol	Pin Type	Min	Typ	Max	Unit
High Output Voltage	V_{OH}	O	—	—	1525	mV
Low Output Voltage	V_{OL}	O	875	—	—	mV
Peak-to-peak Output Differential Voltage	V_{OD}	O	150	—	400	mV
Output Offset Voltage (common-mode voltage)	V_{OS}	O	1075	—	1325	mV
Output Overshoot/Undershoot	V_{RING}	O	—	—	10	%
Output Resistance (single-ended)	R_O	O	40	—	140	Ω
Mismatch of Output Resistance Within a Differential Pair	DR_O	O	—	—	10	%
Change in VOD Between Low and High Output Voltage	D_{VOD}	O	—	—	25	mV
Change in VOS Between Low and High Output Voltage	D_{VOS}	O	—	—	25	mV
Output Short-circuit Current Between an Output and Ground	I_{SA}, I_{SB}	O	—	—	12	mA
Output Short-circuit Current Between the Sides of a Differential Pair	I_{SAB}	O	—	—	10	mA
Powerdown Leakage Current	—	O	—	—	10	mA
Output Rise and Fall Time (20%—80%)	t_R, t_F	O	100	—	200	ps
Differential Skew	Tskew	O	—	—	20	ps

SGMII I/O Receive Specifications

Table 25. SGMII I/O Receive Specifications

Parameter	Symbol	Pin Type	Min	Typ	Max	Unit
Input Voltage Range	V_{ID}	I	675	—	1725	mV
Differential Input Voltage Threshold	V_{IDTH}	I	–50	—	50	mV
Receive Input Differential Resistance	R_I	I	80	—	120	Ω

Electrical Specifications (continued)

SFP—1.25 Gbits/s SerDes Specifications

Table 26. RX Serial Buffer Input Electrical Specification—1.25 Gbits/s SerDes I/O

Parameter	Symbol	Pin Type	Conditions	Min	Typ	Max	Unit
Differential Input Voltage	V_{ID}	I	—	175	—	2000	mV
Common-mode Input Voltage	V_{IC}	I	—	0.5	—	V_{DDIB}	V
Input Resistance	R_I	I	—	—	50	—	Ω
Receiver Rise and Fall Time	t_{RI}, t_{FI}	I	20%—80%	—	—	160	ps
Input Return Loss	S_{11I}	I	—	10	—	—	dB

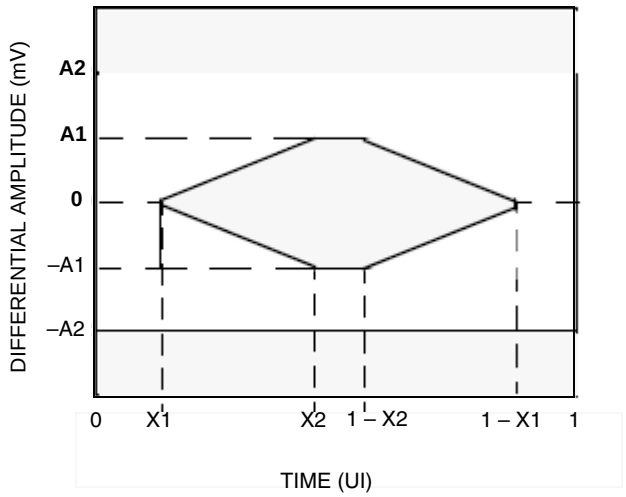


Figure 28. Receiver Eye Diagram

Table 27. Receiver Eye Diagram Values

Symbol	Receiver HDIN P-N Pair	Unit
X1	0.275	UIp*
X2	0.400	UIp
A1	100	mVp
A2	800	mVp
Deterministic Jitter	0.47	UIp-p
Total Jitter	0.65	UIp-p

* UI = 800 ps.

Electrical Specifications (continued)

SFP—1.25 Gbits/s SerDes Specifications (continued)

Table 28. TX Serial Buffer Output Electrical Specification—1.25 Gbits/s SerDes I/O

Parameter	Symbol	Pin Type	Conditions	Min	Typ	Max	Unit
Transmit Differential Voltage Swing	V_{ODAC}	O	All power settings (min—max), ac coupled V_{DDOB} = 1.5	203	—	864	mV
Transmit Differential Voltage Swing	V_{ODDC}	O	All power settings (min—max), direct coupled V_{DDOB} = 1.5	191	—	906	mV
Transmit Differential Voltage Swing	V_{ODAC}	O	All power settings (min—max), ac coupled V_{DDOB} = 1.2	221	—	1134	mV
Transmit Differential Voltage Swing	V_{ODDC}	O	All power settings (min—max), direct coupled V_{DDOB} = 1.2	191	—	1208	mV
Transmit Common-mode Voltage	V_{CMAC}	O	ac coupled	Typ – 0.15	$V_{DDOB/2}$	Typ + 0.15	V
Transmit Common-mode Voltage	V_{CMDC}	O	dc coupled	Typ – 0.15	V_{DDOB} – [V_{OUT}(p-p)/4]	Typ + 0.10	mV
Output Resistance	R_O	O	—	42	50	58	Ω
Transmitter Rise and Fall Time	t_{RO}, t_{FO}	O	20%—80%	60	80	100	ps
Short Circuit Current	I_{OSC}	O	—	—	20	35	mA
Output Return Loss	S_{11O}	O	—	—	10	—	dB

Electrical Specifications (continued)

SFP—1.25 Gbits/s SerDes Specifications (continued)

Table 29. Clocking and Timing Specifications

Parameter	Min	Typ	Max	Unit
Transmitter Output Jitter in Half-Rate Mode:				Ulp-p
Deterministic	—	—	0.08	
Random	—	—	0.12	
Total*	—	—	0.2	

* Does not include in-band reference-clock jitter, which must be added to this. In-band jitter is defined as jitter with spectral content within the 3 dB closed-loop bandwidth of the PPL.

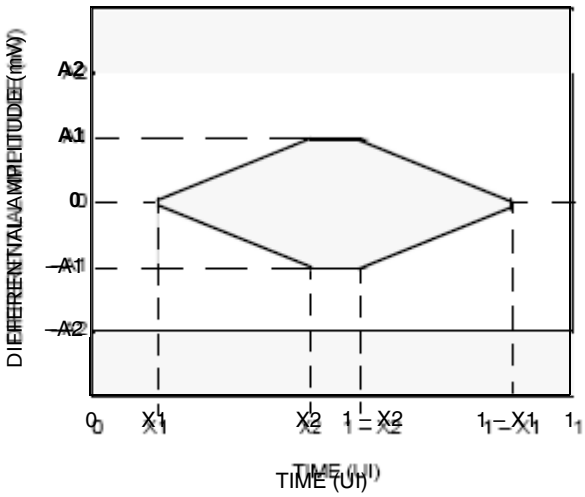


Figure 29. Transmitter Eye Diagram

Table 30. Transmitter Eye Diagram Values

Symbol	Transmitter HDOUT P-N Pair	Unit
X1	0.175	Ulp*
X2	0.390	Ulp
A1	400	mVp
A2	800	mVp
Deterministic Jitter	0.17	Ulp-p
Total Jitter	0.35	Ulp-p

* UI = 800 ps.

Electrical Specifications (continued)

Timing Diagrams

SGMII

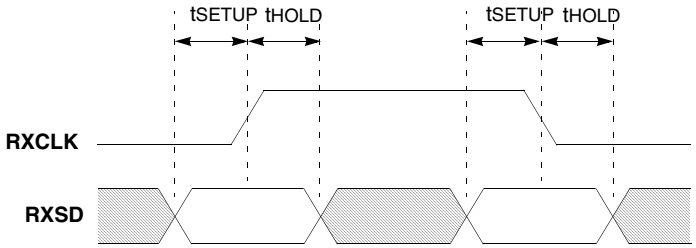


Figure 30. SGMII ac Setup and Hold Timing Diagram

Table 31. SGMII I/O Receive Timing Specification

Parameter	Symbol	Min	Typ	Max	Unit
Setup Time	tSETUP	100	—	—	ps
Hold Time	tHOLD	100	—	—	ps

PCI

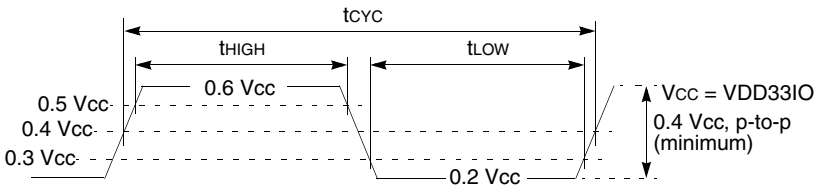


Figure 31. 3.3 V *PCI* Clock Waveform

Table 32. 3.3 V *PCI* Clock ac Specification

Parameter	Symbol	Min	Typ	Max	Unit
Clock Cycle Time	tCYC	15	—	30	ns
Clock High Time	tHIGH	6	—	—	ns
Clock Low Time	tLOW	6	—	—	ns
Clock Slew Rate	—	1	—	4	V/ns
Rise and Fall Time*	tR, tF	1.8	—	5.84	ns

* Test conditions, 30%—60%, 10 pF load plus pad and package C, 140 Ω to V_{DD33_IO}, and 140 Ω to V_{SS}.

Electrical Specifications (continued)

Timing Diagrams (continued)

PCI (continued)

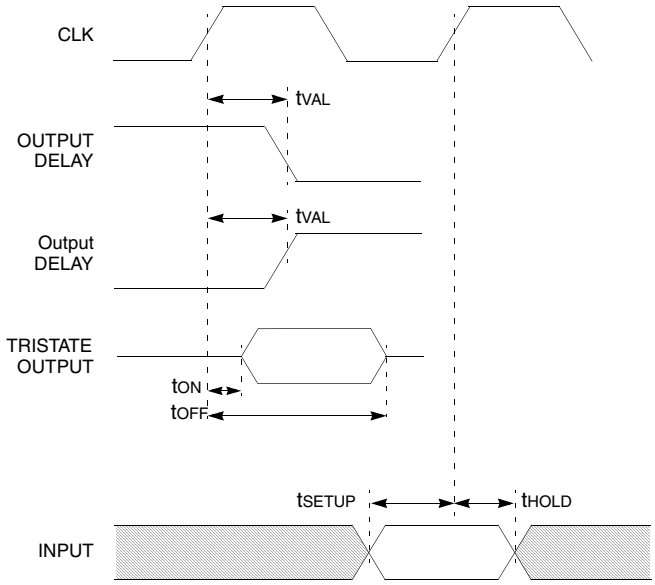


Figure 32. *PCI* Timing Diagram

Table 33. 3.3 V *PCI* Clock ac Specification

Parameter	Symbol	Min	Typ	Max	Unit
Clock-to-signal Valid	tVAL	2	—	6	ns
Float-to-active Delay	tON	2	—	—	ns
Active-to-float Delay	tOFF	—	—	14	ns
Setup Time	tSETUP	3	—	—	ns
Hold Time	tHOLD	0	—	—	ns

Electrical Specifications (continued)

Timing Diagrams (continued)

SerDes I/O Clock Input Specification

Table 34. SerDes Reference Clock Specifications

Parameter	Symbol	Min	Typ	Max	Unit
Clock Frequency—SFP (REFCLK_3)	—	—	125.0	—	MHz
Frequency Stability	—	–100		100	ppm
Duty Cycle	—	40	50	60	%
Rise Time and Fall Time (20%—80%)	t_R, t_F		—	1.2	ns
Differential Amplitude	V_{OD}	600	—	2 x VDD_3	mVp-p
Single-ended Amplitude	V_{SE}	300	—	VDD_3	mVp-p
Common-mode Level	VCM	$V_{SE}/2$	—	VDD_3 – $V_{SE}/2$	V
PLL Bandwidth	—	—	—	12	MHz
Input Capacitance	CI	—	—	2.0	pF
Skew, REFCLK_3_(P,N)	—	–75	—	75	ps
Jitter (peak-to-peak)	—	—	—	100	ps_{p-p}

Core Clock Input Specifications

This pertains to signals named REFCLK_CORE and the supporting PLL voltage pins, AVDD_PLL_CORE and AGND_PLL_CORE.

Table 35. Core Clock Input Specifications

Parameter	Symbol	Min	Typ	Max	Unit
Maximum Input Voltage	$V_{IH,MAX}$	—	—	5.5	V
Minimum Input Voltage	$V_{IL,MIN}$	–0.5	—	—	V
Input High Voltage	V_{IH}	1.3	1.55	2.0	V
Input Low Voltage	V_{IL}	0.8	1.20	1.5	V
Input Hysteresis	V_{HYS}	0.3	0.35	0.4	V
Input Frequency	F_{CORE}	—	25.000	—	MHz
Input Frequency Stability	—	–100	—	100	ppm
Rise Time and Fall Time (20%—80%)	t_R, t_F	—	—	1.2	ns
Jitter (peak-to-peak)	—	—	—	100	ps_{p-p}

Electrical Specifications (continued)

Timing Diagrams (continued)

SGMII PLL Timing and Clocking Specification

Table 36. SGMII PLL Timing and Clocking Specification

Parameter	Symbol	Pin Type	Min	Typ	Max	Unit
Maximum Input Voltage	$V_{IH,MAX}$	—	—	—	5.5	V
Minimum Input Voltage	$V_{IL,MIN}$	—	–0.5	—	—	V
Input High Voltage	V_{IH}	—	1.3	1.55	2.0	V
Input Low Voltage	V_{IL}	—	0.8	1.20	1.5	V
Input Hysteresis	V_{HYS}	—	0.3	0.35	0.4	V
Clock Duty Cycle	t_{DC}	I	48	—	5.5	%
Clock Jitter	t_{JCC}	I	—	—	50	ps
Input Frequency	F	I	—	25	—	MHz
Input Frequency Stability	—	I	–100	—	100	ppm

Serial Management Interface Timing

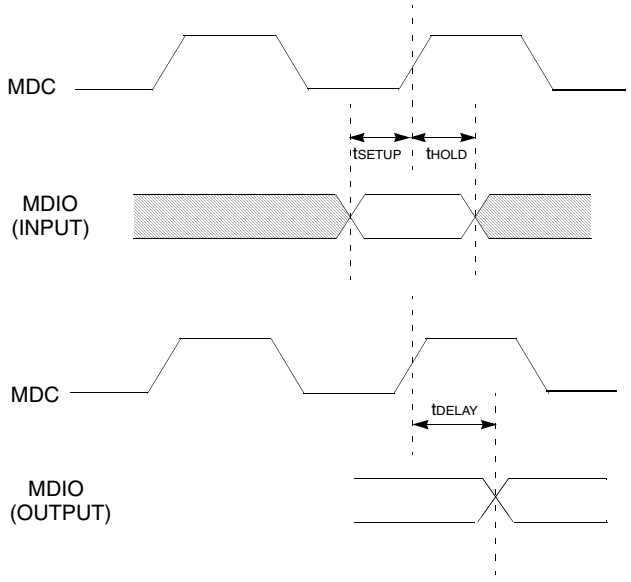


Figure 33. Serial Management Interface Timing

Table 37. Serial Management Interface Timing

Parameter	Min	Typ	Max	Unit
MDIO Setup to MDC	10	—	—	ns
MDIO Hold to MDC	10	—	—	ns
MDC to MDIO Delay	—	—	20	ns

Electrical Specifications (continued)

Timing Diagrams (continued)

JTAG Timing

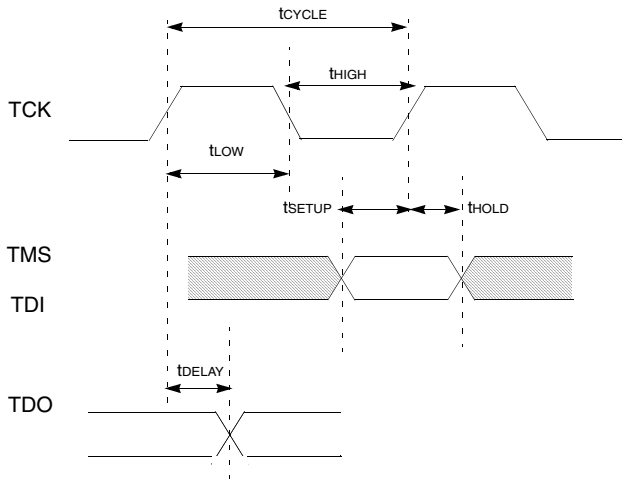
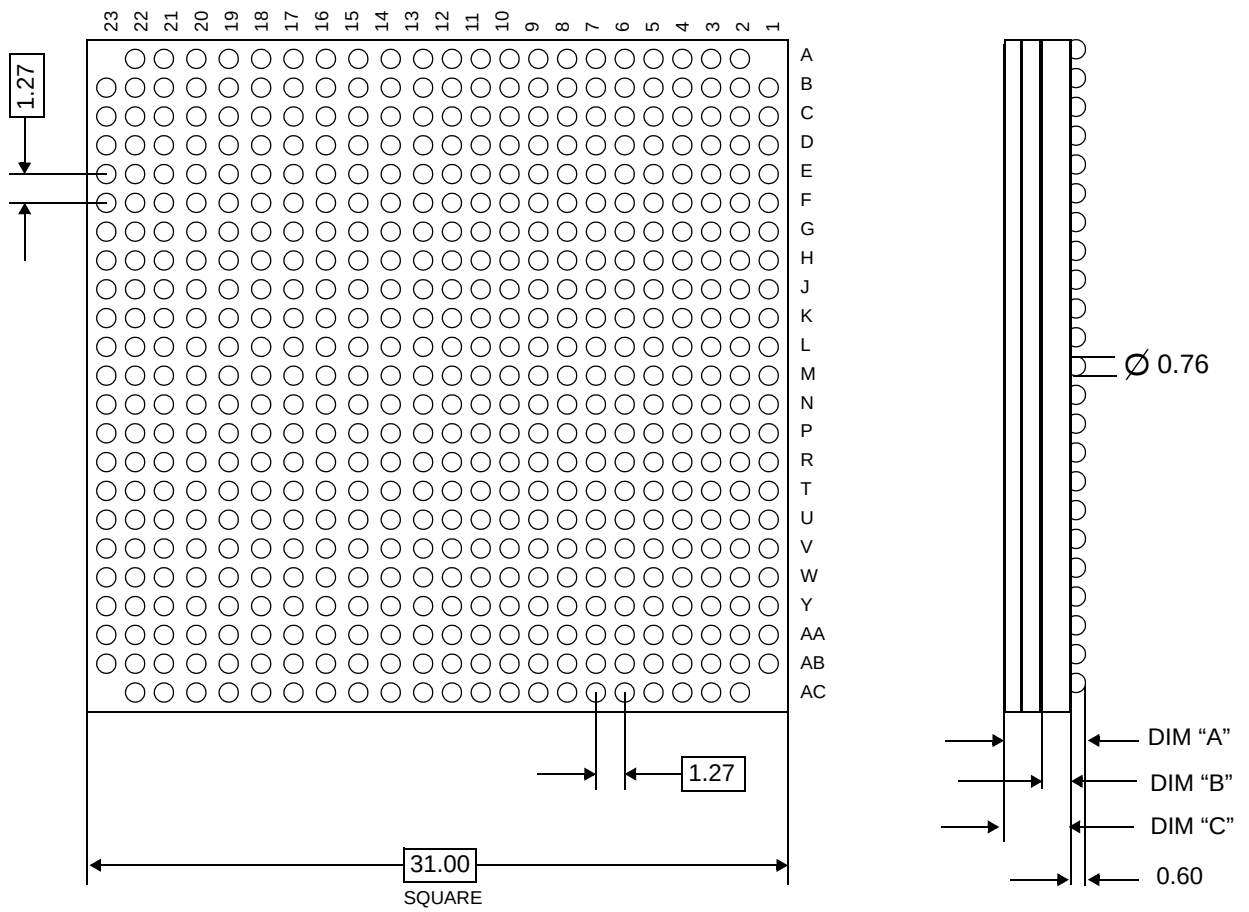


Figure 34. JTAG Timing

Table 38. JTAG Timing

Parameter	Min	Typ	Max	Unit
TCK Period	40	—	—	ns
TCK High	20	—	—	ns
TCK Low	20	—	—	ns
TDI, TMS to TCK Setup	15	—	—	ns
TDI, TMS to TCK Hold	15	—	—	ns
TCK to TDO Delay	0	—	30	ns

Physical Dimensions



B, B1	3.22 MAX	1.08 NOM	2.46 ± 0.30
C	3.12 MAX	0.99 NOM	2.36 ± 0.30
Revision	**DIM “A”**	**DIM “B”**	**DIM “C”**
THICKNESS SCHEDULE			

Figure 35. ET3028-50 Physical Dimensions

Appendix A: Registers

This chapter contains detailed descriptions of all of the registers embodied within the ET3028-50. The register definitions are arranged alphabetically for easy reference.

Each definition consists of a register parameters table, a register diagram, and a field parameters table.

The register parameters table defines the parameters that apply to the registers as a whole. These parameters include such items as a register's base address and size.

The register diagram is a graphical depiction of the layout of the register's fields. Where space permits, the name of the field is inserted into its position in the diagram. Unused bits are represented by gray blocks. These unused areas return zeros when read and are nonreactive to writes.

The field parameters table provides parameters and descriptions for all of the various fields that may make up a particular register.

Registers, Records, and Fields

A hierarchy exists in the arrangement and presentation of fields within a register. Fields are grouped together into records. Records are grouped together into registers. Multiple instances of any field, record, or register may be implemented. When multiple instances do exists, the spacing between the instances is specified. The spacing parameter counts bits from the leftmost bit of one instance to the leftmost bit of the next instance.

Offsets are measured from the beginning of a record. This measurement is depicted in a bytes.bits format. For example, the value 8.3 means 8 bytes and 3 bits.

Instance Numbering

Instance numbering ranges are depicted by a pair of numbers within curly braces, for example: {0..5}. These ranges correspond to instance parameters. For the present example, this range corresponds to an instances parameter of 6.

When multiple range values are present in a field name, it implies that multiple nested instance numbering ranges apply. The hierarchy: register -> record -> field is used. An empty set of braces indicates that the corresponding instance numbering range does not appear in the object's name.

The end result of this numbering method is that the actual names of the objects contain numbers from within the specified ranges and do not include any curly braces.

Line Caching

The records for certain tables are wider than the supervisor's 32-bit data bus, and it is imperative that records be updated in a single operation in order to ensure reliable table operation. Therefore, the ET3028-50 includes an automatic line cache for use with these registers. Proper operation of the line cache depends upon the data being written to a register's wide record in a particular order: from offset zero to the last 32-bit word of the record. When the supervisor performs a write to the last offset of the record, the line cache is automatically written to the desired register record as a single, wide word.

Appendix A: Registers (continued)

Acl_Deny_Packets

Description: The number of packets denied access by the ACL function.

Table 39. Acl_Deny_Packets Register Parameters

Parameter	Value
Base Address	0x0004_2508
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

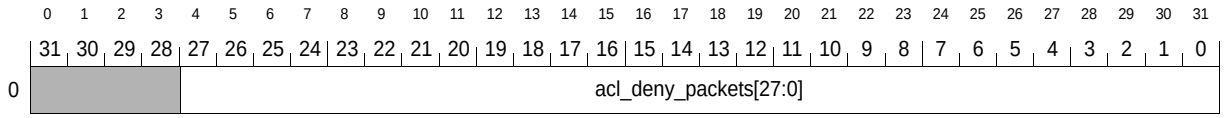


Figure 36. Acl_Deny_Packets Diagram

Table 40. Acl_Deny_Packets Field Parameters

Field Name	Parameters	Description
acl_deny_packets[27:0]	Mode = R/W Offset = 0.4 Instances = 1	This counter is incremented each time a packet is denied by the ACL function. This counter does not stick at its maximum value, nor is any indication of a rollover provided to the supervisor. Therefore, the supervisor must sample this register often enough to prevent an undetected rollover.

Appendix A: Registers (continued)

Acl_En

Description: Enables the individual ACLs.

Table 41. Acl_En Register Parameters

Parameter	Value
Base Address	0x0004_2500
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

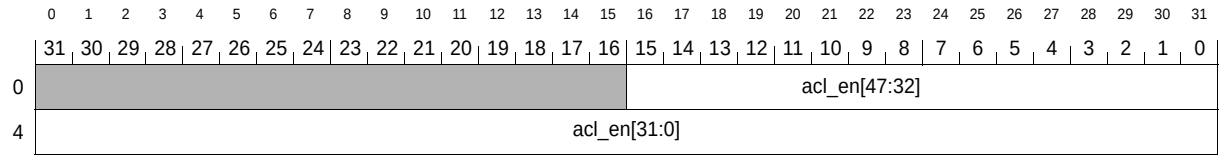


Figure 37. Acl_En Register Diagram

Table 42. Acl_En Field Parameters

Field Name	Parameters	Description
acl_en[47:0]	Mode = R/W Offset = 0.16 Instances = 1	Each bit of this field corresponds to an ACL. An ACL is enabled when its bit is asserted. If an ACL is disabled, it is considered to be empty, and a permit action is implied.

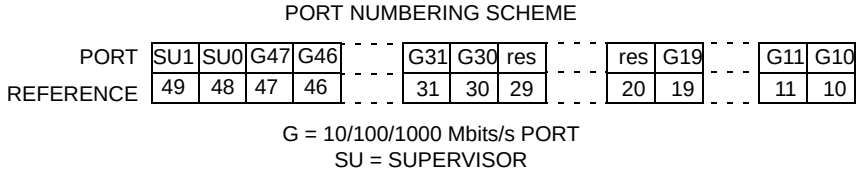


Figure 38. Port Numbering Scheme

Appendix A: Registers (continued)

Acl_Ip_Addr_Ace_Map_Index_Table

Description: This table converts an IP address index into an ACE map index.

Table 43. Acl_Ip_Addr_Ace_Map_Index_Table Register Parameters

Parameter	Value
Base Address	0x0000_0000
Register Size	16384
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	4K
Record Spacing	4

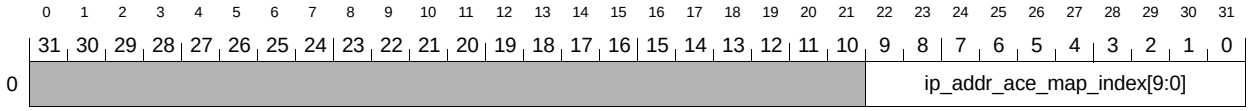


Figure 39. Acl_Ip_Addr_Ace_Map_Index_Table Register Diagram

Table 44. Acl_Ip_Addr_Ace_Map_Index_Table Field Parameter

Field Name	Parameters	Description
ip_addr_ace_map_index[9:0]	Mode = R/W Offset = 0.22	The ACE map index value.

This table is addressed by the concatenation of the `acl_index[5:0]` value and the IP address look-up result (`ip_src_addr_index[5:0]` or `ip_dest_addr_index[5:0]`). `acl_index[5:0]` occupies the upper portion of the concatenated address word. Three bits of zero are placed between acl_index[5:0] and the IP address look-up result such that each increment of `acl_index[5:0]` advances the pointer into `Acl_IP_Addr_Ace_Map_Index_Table` by 512 entries. Each entry in this table is a 10-bit index into `Acl_Ip_Addr_Ace_Map_Table`.

This table allows the 64 IP addresses appearing in 48 ACLs to address as many as 1,024 ACE maps. The following figure shows where this table fits in the processing pipeline.

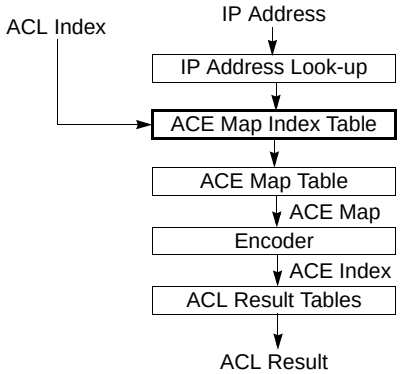


Figure 40. ACL Processing Pipeline

Appendix A: Registers (continued)

Acl_Ip_Addr_Ace_Map_Table

Description: This table converts ACE map index values into ACE maps.

Table 45. Acl_Ip_Addr_Ace_Map_Table System Parameters

Parameter	Value
Base Address	0x0003_c000
Register Size	8192
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1K
Record Spacing	8

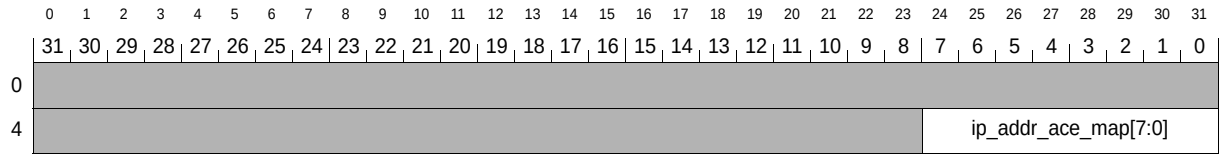


Figure 41. Acl_Ip_Addr_Ace_Map_Table Register Diagram

Table 46. Acl_Ip_Addr_Ace_Map_Table Field Parameters

Field Name	Parameters	Description
ip_addr_ace_map[7:0]	Mode = R/W Offset = 4.24	The ACE map value.

This table is addressed by `Acl_Ip_Addr_Ace_Map_Index_Table.ip_addr_ace_map_index[9:0]` and returns the 8-bit ACE map value for the associated IP address. An ACE map is a vector that identifies all of the ACEs for which the associated IP address generates a match.

The following figure shows where this table fits in the ACL processing pipeline.

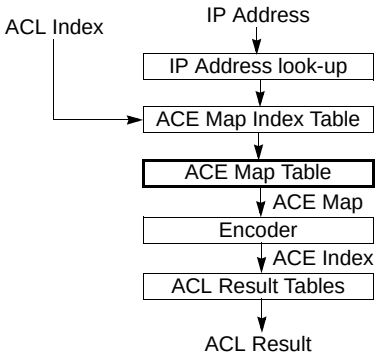


Figure 42. ACL Processing Pipeline

Appendix A: Registers (continued)

Acl_Ip_Key_Table_0

Description: This table is reserved and must be programmed to all ones for proper ACL operation.

Table 47. Acl_Ip_Key_Table_0 Register Parameters

Parameter	Value
Base Address	0x0004_24e0
Register Size	16
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	1
Record Spacing	NA

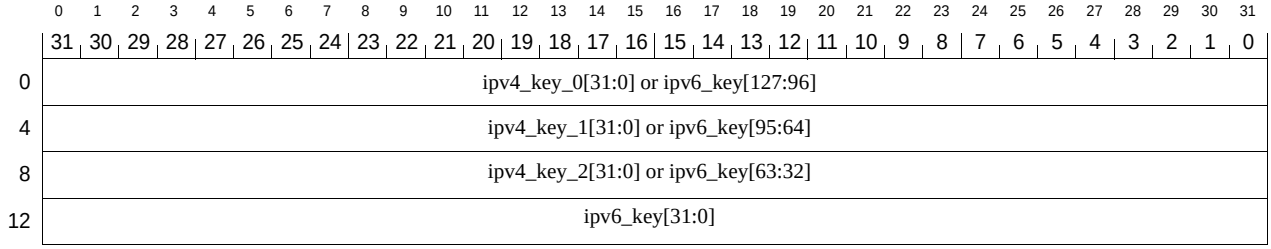


Figure 43. Acl_Ip_Key_Table_0 Register Diagram

Table 48. Acl_Ip_Key_Table_0 Field Parameters

Field Name	Parameters	Description
ipv4_key_{0..2}[31:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 4.0	All fields must be filled with all ones for proper operation; (i.e., program ipv4_key_0[31:0] to 0xFFFFFFFF, ipv4_key_1[31:0] to 0xFFFFFFFF, and ipv4_key_2[31:0] to 0xFFFFFFFF.)
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	All fields must be filled with all ones for proper operation; (i.e., program ipv6_key[127:96] to 0xFFFFFFFF, ipv6_key[95:64] to 0xFFFFFFFF, ipv6_key[63:32] to 0xFFFFFFFF, and ipv6_key[31:0] to 0xFFFFFFFF.)

This register is reserved and is a placeholder for future expansion. All fields must be filled with all ones for proper operation.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_1

Description: This table is reserved and must be programmed to all ones for proper ACL operation.

Table 49. Acl_Ip_Key_Table_1 Register Parameters

Parameter	Value
Base Address	0x0004_2480
Register Size	16
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	1
Record Spacing	16

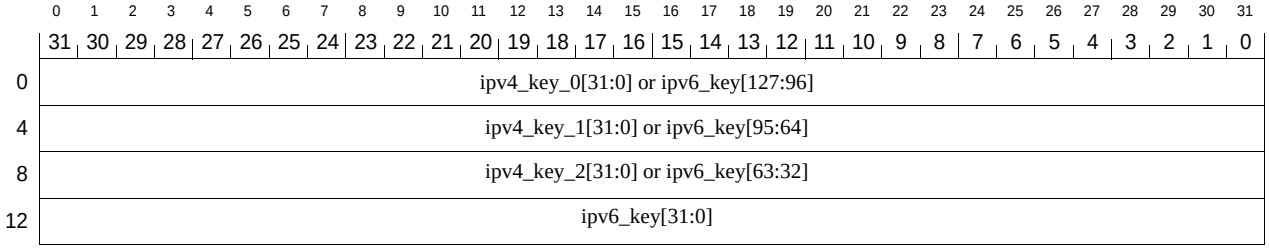


Figure 44. Acl_Ip_Key_Table_1 Register Diagram

Table 50. Acl_Ip_Key_Table_1 Field Parameters

Field Name	Parameters	Description
ipv4_key_{0..2}[31:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 4.0	All fields must be filled with all ones for proper operation; (i.e., program ipv4_key_0[31:0] to 0xFFFFFFFF, ipv4_key_1[31:0] to 0xFFFFFFFF, and ipv4_key_2[31:0] to 0xFFFFFFFF.)
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	All fields must be filled with all ones for proper operation; (i.e., program ipv6_key[127:96] to 0xFFFFFFFF, ipv6_key[95:64] to 0xFFFFFFFF, ipv6_key[63:32] to 0xFFFFFFFF, and ipv6_key[31:0] to 0xFFFFFFFF.)

This register is reserved and is a placeholder for future expansion. All fields must be filled with all ones for proper operation.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_2

Description: This table performs the first stage of IPv4-only address look-up.

Table 51. Acl_Ip_Key_Table_2 Register Parameters

Parameter	Value
Base Address	0x0004_2300
Register Size	16
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	1
Record spacing	16

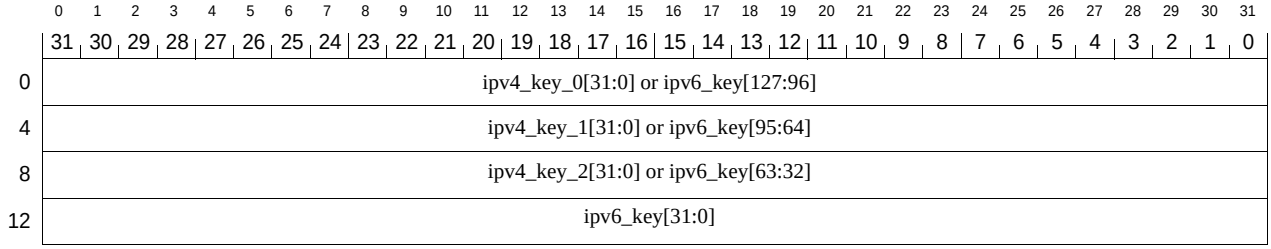


Figure 45. Acl_Ip_Key_Table_2 Register Diagram

Table 52. Acl_Ip_Key_Table_2 Field Parameters

Field Name	Parameters	Description
ipv4_key_{0..2}[31:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 4.0	A set of 32-bit IPv4 address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	All fields must be filled with all ones for proper operation; (i.e., program ipv6_key[127:96] to 0xFFFFFFFF, ipv6_key[95:64] to 0xFFFFFFFF, ipv6_key[63:32] to 0xFFFFFFFF, and ipv6_key[31:0] to 0xFFFFFFFF.)

For IPv4-only, this first stage of IP address look-up utilizes just a single record. Three IPv4 address keys are stored here. The comparisons result in an index computation. For IPv4-only, one of four index values is chosen. These index values are used to address records in the next stage of the look-up.

This register is reserved for IPv4/IPv6 and is a placeholder for future expansion. For IPv4/IPv6, all fields must be filled with all ones for proper operation.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_3

Description: This table performs the second stage of IPv4-only address look-up.

Table 53. Acl_Ip_Key_Table_3 Register Parameters

Parameter	Value
Base Address	0x0004_1c00
Register Size	64
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	4
Record Spacing	16

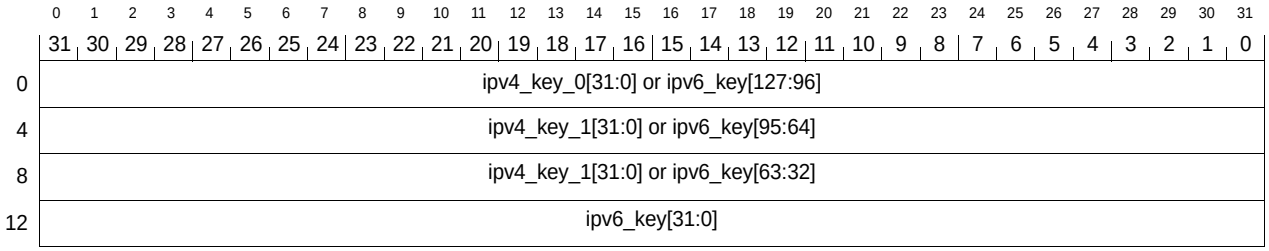


Figure 46. Acl_Ip_Key_Table_3 Register Diagram

Table 54. Acl_Ip_Key_Table_3 Field Parameters

Field Name	Parameters	Description
ipv4_key_{0..2}[31:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 4.0	A set of 32-bit IPv4 address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	All fields must be filled with all ones for proper operation; (i.e., program ipv6_key[127:96] to 0xFFFFFFFF, ipv6_key[95:64] to 0xFFFFFFFF, ipv6_key[63:32] to 0xFFFFFFFF, and ipv6_key[31:0] to 0xFFFFFFFF.)

For IPv4-only, this second stage of IP address look-up utilizes four records.

This register is reserved for IPv4/IPv6 and is a placeholder for future expansion. For IPv4/IPv6, all fields must be filled with all ones for proper operation.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_4

Description: This table performs the third stage of IPv4-only address look-up and the first stage of IPv4/IPv6 address look-up.

Table 55. Acl_Ip_Key_Table_4 Register Parameters

Parameter	Value
Base Address	0x0003_e000
Register Size	256
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	16
Record Spacing	16

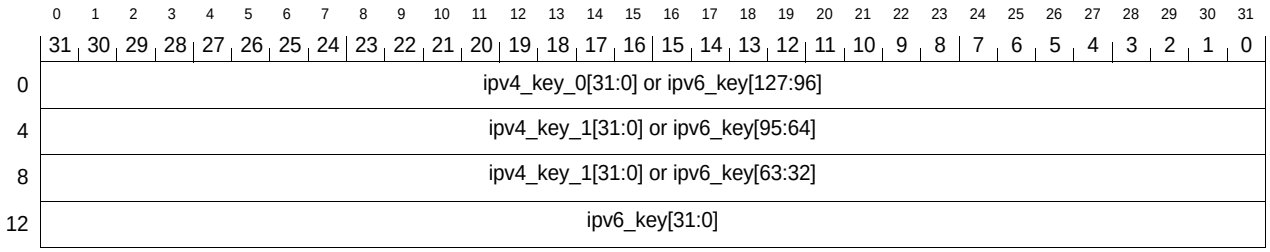


Figure 47. Acl_Ip_Key_Table_4 Register Diagram

Table 56. Acl_Ip_Key_Table_4 Field Parameters

Field Name	Parameters	Description
ipv4_key_{0..2}[31:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 4.0	A set of 32-bit IPv4 address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	A 128-bit IPv6 address value. This value is compared against the search argument. The result of this comparison is used to select one of two index values for use in the next stage of the look-up.

For IPv4-only, this third stage of IP address look-up utilizes 16 records.

For IPv4/IPv6, this first stage of IP address look-up utilizes just a single record. One IPv6 address key is stored here. The comparisons result in an index computation. For IPv4/IPv6, one of two index values is chosen. These index values are used to address records in the next stage of the look-up.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_5

Description: This table performs the fourth and final stage of IPv4-only and the second stage of IPv4/IPv6 address look-up.

Table 57. Acl_Ip_Key_Table_5 Register Parameters

Parameter	Value
Base Address	0x0003_8000
Register Size	1024
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	64
Record Spacing	16

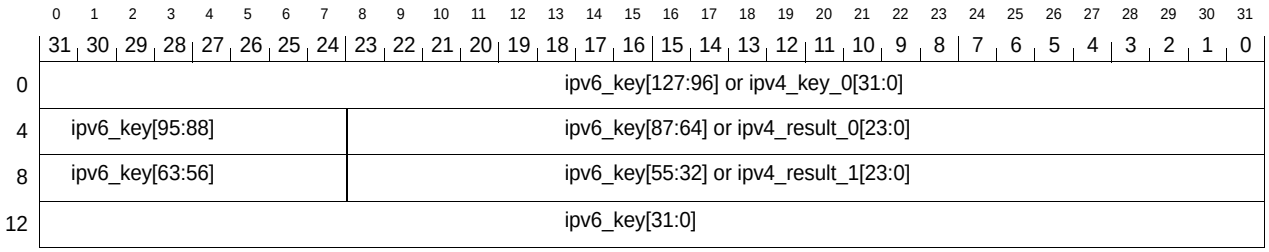


Figure 48. Acl_Ip_Key_Table_5 Register Diagram

Appendix A: Registers (continued)

Acl_Ip_Key_Table_5 (continued)

Table 58. Acl_Ip_Key_Table_5 Field Parameters

Field Name	Parameters	Description
ipv4_key[31:0]	Mode = R/W Offset = 0.0 Instances = 1	A 32-bit IPv4 address value. This value is compared against the search argument. The result of this comparison is used to select one of two result values.
ipv4_result_{0..1}[23:0]	Mode = R/W Offset = 4.8 Instances = 2 Spacing = 4.0	One of these 24-bit results is returned as the final result of an IPv4 address look-up. `ip_src_addr_index_{0..1}[5:0]` and `ip_dest_addr_index_{0..1}[5:0]` are used in conjunction with the ACL index to select one of 1,024 ACE maps for the corresponding IP source and destination addresses, respectively. `ip_src_addr_flow_id_{0..1}[2:0]` and `ip_dest_addr_flow_id_{0..1}[2:0]` identify flows associated with the matching IP source and destination addresses, respectively.
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	A 128-bit IPv6 address value. This value is compared against the search argument. The result of this comparison is used to select one of two index values for use in the next stage of the look-up.

For IPv4-only, this fourth stage of IP address look-up marks the end of processing for IPv4 addresses. Each record contains two possible sets of result values for IPv4-only.This stage utilizes 2 records for IPv4/IPv6 or 64 records for IPv4-only.

`ipv4_result_{0..1}[23:0]` utilizes the following format:

```
{ reserved[2:0], ip_src_addr_index_{0..1}[5:0], ip_src_addr_flow_id_{0..1}[2:0],
reserved[2:0],
ip_dest_addr_index_{0..1}[5:0], ip_dest_addr_flow_id_{0..1}[2:0] }
```

Appendix A: Registers (continued)

Acl_Ip_Key_Table_6

Description: This table performs the third stage of IPv4/IPv6 address look-up.

Table 59. Acl_Ip_Key_Table_6 Register Parameters

Parameter	Value
Base Address	0x0004_1800
Register Size	64
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	4
Record Spacing	16

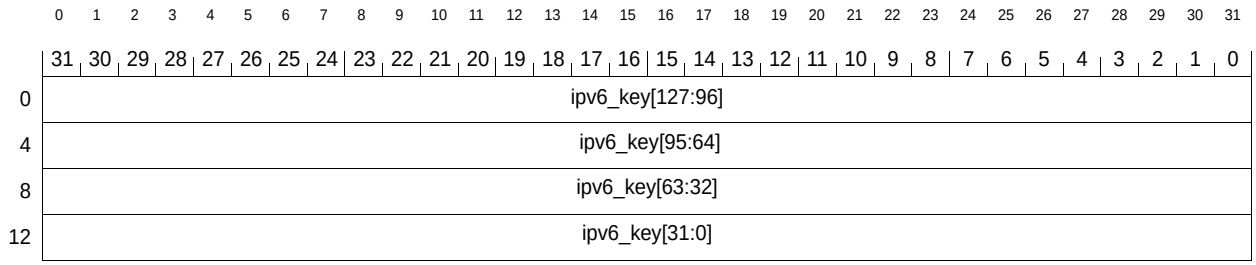


Figure 49. Acl_Ip_Key_Table_6 Register Diagram

Table 60. Acl_Ip_Key_Table_6 Field Parameters

Field Name	Parameters	Description
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	A 128-bit IPv6 address value. This value is compared against the search argument. The result of this comparison is used to select one of two index values for use in the next stage of the look-up.

This stage of IP address look-up operates only on IPv4/IPv6 addresses. This stage utilizes four records for IPv4/IPv6.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_7

Description: This table performs the fourth stage of IPv4/IPv6 address look-up.

Table 61. Acl_Ip_Key_Table_7 Register Parameters

Parameter	Value
Base Address	0x0004_0800
Register Size	128
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	8
Record Spacing	16

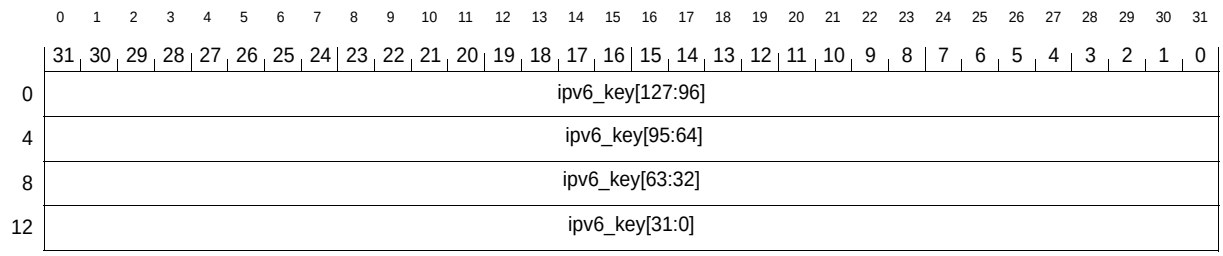


Figure 50. Acl_Ip_Key_Table_7 Register Diagram

Table 62. Acl_Ip_Key_Table_7 Field Parameters

Field Name	Parameters	Description
ipv6_key[127:0]	Mode = R/W Offset = 0.0 Instances = 1	A 128-bit IPv6 address value. This value is compared against the search argument. The result of this comparison is used to select one of two index values for use in the next stage of the look-up.

This stage of IP address look-up operates only on IPv6 addresses. This stage utilizes eight records for IPv6.

Appendix A: Registers (continued)

Acl_Ip_Key_Table_8

Description: This table performs the fifth and final stage of IP address look-up.

Table 63. Acl_Ip_Key_Table_8 Register Parameters

Parameter	Value
Base Address	0x0003_0000
Register Size	1024
Register Instances	1
Register Spacing	NA
Record Size	44
Record Instances	16
Record Spacing	64

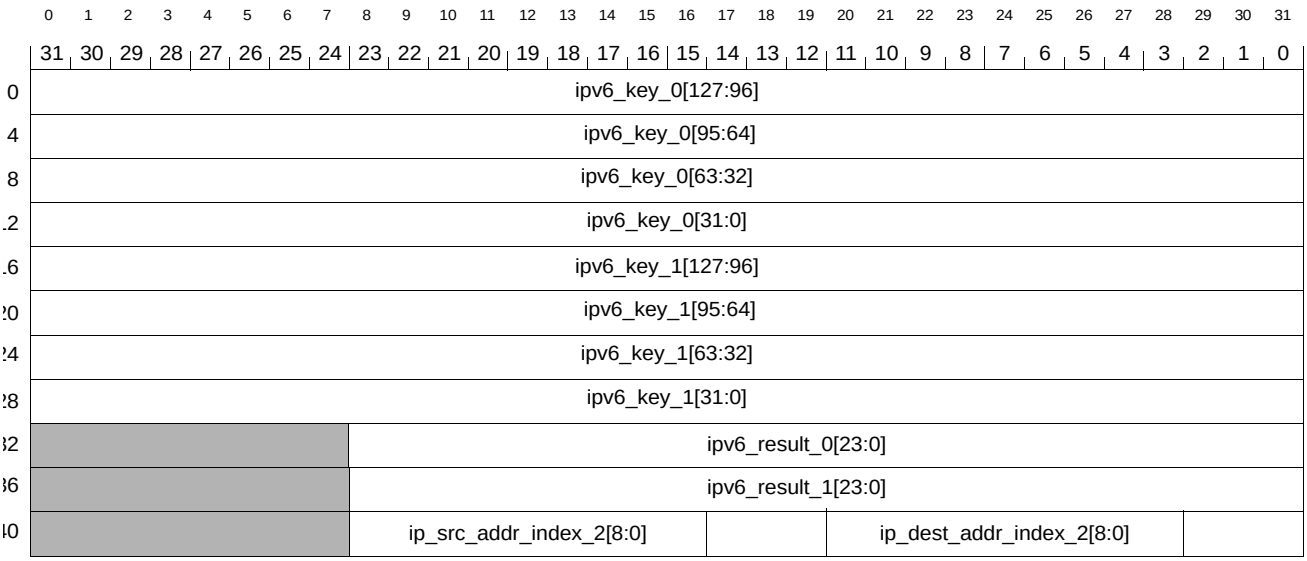


Figure 51. Acl_Ip_Key_Table_8 Register Diagram

Appendix A: Registers (continued)

Acl_Ip_Key_Table_8 (continued)

Table 64. Acl_Ip_Key_Table_8 Field Parameters

Field Name	Parameters	Description
ipv6_key_0[127:0]	Mode = R/W Offset = 0.0 Instances = 1	A 128-bit IPv6 address value. This value is compared against the search argument. The result of this comparison is used to select one of two result values.
ipv6_key_1[127:0]	Mode = R/W Offset = 16.0 Instances = 1	All fields must be filled with all ones for proper operation; (i.e. program ipv6_key_1[127:96] to 0xFFFFFFFF, ipv6_key_1[95:64] to 0xFFFFFFFF, ipv6_key_1[63:32] to 0xFFFFFFFF, and ipv6_key_1[31:0] to 0xFFFFFFFF.)
ipv6_result_{0..1}[23:0]	Mode = R/W Offset = 32.8 Instances = 2 Spacing = 4.0	One of these 24-bit results is returned as the final result of an IPv6 address look-up. ip_src_addr_index_{0..1}[5:0] and ip_dest_addr_index_{0..1}[5:0] are used in conjunction with the ACL index to select one of 1,024 ACE maps for the corresponding IP source and destination addresses, respectively. ip_src_addr_flow_id_{0..1}[2:0] and ip_dest_addr_flow_id_{0..1}[2:0] identify a flow associated with the matching IP source and destination addresses, respectively.
ip_src_addr_index_{2}[8:0]	Mode = R/W Offset = 40.8 Instances = 1	This field must be filled with all ones for proper operation; (i.e. program ip_src_addr_index_{2}[8:0] to 0x1FF.)
ip_src_addr_flow_id_{2}[2:0]	Mode = R/W Offset = 40.17 Instances = 1	This field must be filled with all ones for proper operation; (i.e. program ip_src_addr_flow_id_{2}[2:0] to 0x7.)
ip_dest_addr_index_{2}[8:0]	Mode = R/W Offset = 40.20 Instances = 1	This field must be filled with all ones for proper operation; (i.e. program ip_dest_addr_index_{2}[8:0] to 0x1FF.)
ip_dest_addr_flow_id_{2}[2:0]	Mode = R/W Offset = 40.29 Instances = 1	This field must be filled with all ones for proper operation; (i.e. program ip_dest_addr_flow_id_{2}[2:0] to 0x7.)

This final stage of IP address look-up operates only on IPv6 addresses. This stage utilizes 16 records with one 128-bit IPv6 address value each. This value is compared against the search argument. The result of this comparison is used to select one of two result values.

ipv6_result_{0..1}[23:0] utilizes the following format:

```
{ reserved[2:0], ip_src_addr_index_{0..1}[5:0],
ip_src_addr_flow_id_{0..1}[2:0], reserved[2:0],
ip_dest_addr_index_{0..1}[5:0], ip_dest_addr_flow_id_{0..1}[2:0] }
```

Appendix A: Registers (continued)

Acl_Mode

Description: Global operating modes for the ACL look-up function.

Table 65. Acl_Mode Register Parameters

Parameter	Value
Base Address	0x0004_24f8
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

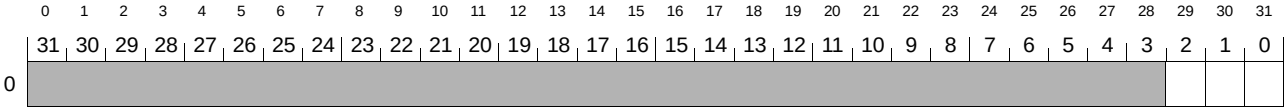


Figure 52. Acl_Mode Register Diagram

Table 66. Acl_Mode Register Field Parameters

Field Name	Parameters	Description
ipv4_only	Mode = R/W Offset = 0.29 Instances = 1 Reset = 0	When asserted, this bit implies that only 32-bit IPv4 address values may be accommodated by the IP address tables. When deasserted, a mix of 128-bit IPv6 and zero-prefixed, right-justified 32-bit IPv4 addresses may be used. The IPv4-only table capacity is 64 addresses. For a mix of IPv4 and IPv6 addresses, the capacity drops to 16.
port_based_acls	Mode = R/W Offset = 0.30 Instances = 1 Reset = 0	This bit is asserted to direct the system to use a packet's receive port to identify its ACL. If this bit is deasserted, then the receive packet's VLAN index is used instead. **Note**: The maximum number of VLANs supported is 256 yet the maximum number of ACLs is 28. `Acl_Vlan_Index_Table` is used to map VLAN indexes to ACL numbers.
auto_deny_log_en	Mode = R/W Offset = 0.31 Instances = 1 Reset = 0	If a packet is denied access because a matching ACE was not found, then the packet is logged if this bit is asserted. If deasserted, then a packet that fails to match any ACE is simply discarded.

This register provides a series of general configuration and mode bits for the ACL look-up function.

Appendix A: Registers (continued)

Acl_Port_Ace_Map_Index_Table

Description: This table converts a port number range index into an ACE map index.

Table 67. Acl_Port_Ace_Map_Index_Table Register Parameters

Parameter	Value
Base Address	0x0002_0000
Register Size	3072
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	768
Record Spacing	4

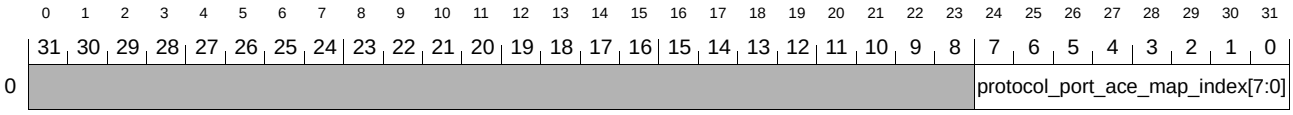


Figure 53. Acl_Port_Ace_Map_Index_Table Register Diagram

Table 68. Acl_Port_Ace_Map_Index_Table Field Parameters

Field Name	Parameters	Description
protocol_port_ace_map_index[7:0]	Mode = R/W Offset = 0.24	The ACE map index value.

This table is addressed by the concatenation of the `acl_index[5:0]` value and the TCP port look-up result `(tcp_src_port_index[3:0] or tcp_dest_port_index[3:0])`. `acl_index[5:0]` occupies the upper portion of the concatenated address word. Four bits of zero are placed between acl_index[5:0] and the TCP port look-up result such that each increment of `acl_index[5:0]` advances the pointer into `Acl_Port_Ace_Map_Index_Table` by 256 entries. Each entry in this table is an 8-bit index into `Acl_Protocol_Port_Ace_Map_Table`.

This table allows the 16 TCP port ranges appearing in 48 ACLs to address as many as 256 ACE maps. The following figure shows where this table fits in the processing pipeline.

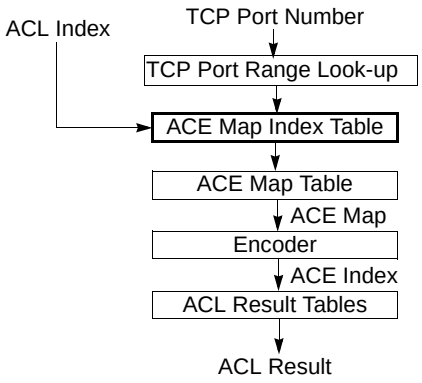


Figure 54. ACL Processing Pipeline

Appendix A: Registers (continued)

Acl_Port_Ace_Map_Table

Description: This table converts ACE map index values derived from TCP port indexes into ACE maps.

Table 69. Acl_Port_Ace_Map_Table Register Parameters

Parameter	Value
Base Address	0x0004_0000
Register Size	2048
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	256
Record Spacing	8

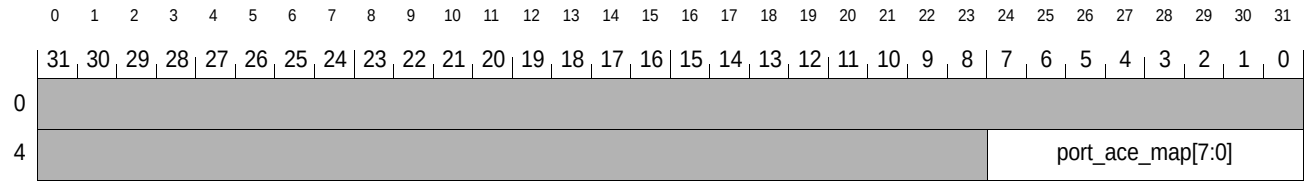


Figure 55. Acl_Port_Ace_Map_Table Register Diagram

Table 70. Acl_Port_Ace_Map_Table Field Parameters

Field Name	Parameters	Description
port_ace_map[63:0]	Mode = R/W Offset = 0.0	The ACE map value.

This table is addressed by `Acl_Port_Ace_Map_Index_Table.port_ace_map_index[7:0]` and returns the 8-bit ACE map value for the associated TCP port. An ACE map is a vector that identifies all of the ACEs for which the associated TCP port number range registers a match.

The following figure shows where this table fits in the ACL processing pipeline.

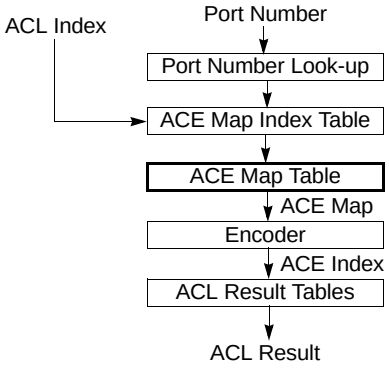


Figure 56. ACL Processing Pipeline

Appendix A: Registers (continued)

Acl_Priority_Update_En

Description: Per-port priority update enables.

Table 71. Acl_Priority_Update_En Register Parameters

Parameter	Value
Base Address	0x0004_2510
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

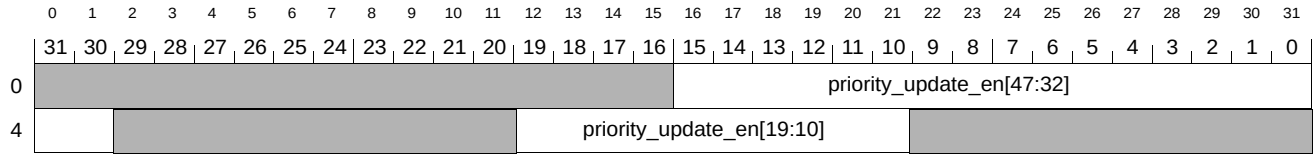


Figure 57. Acl_Priority_Update_En Register Diagram

Table 72. Acl_Priority_Update_En Field Parameters

Field Name	Parameters	Description
priority_update_en[47:10]	Mode = R/W Offset = 0.16 Instances = 1 Reset = 0	When a bit is asserted in this field, the ACL function is enabled to replace the packet's parser-derived priority value with one that is determined by ACL classification. Bits 10 through 19 and 30 through 47 correspond to the 1 Gbit/s ports.

This register provides a per-port method for enabling the classification-based replacement of packet priority values.

.

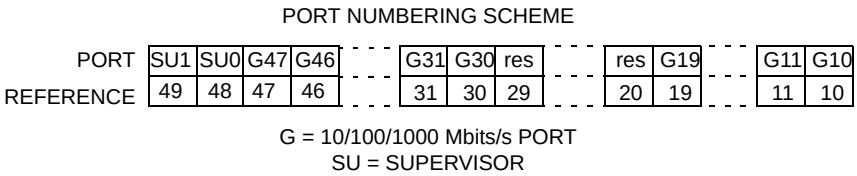


Figure 58. Port Numbering Scheme

Appendix A: Registers (continued)

Acl_Protocol_Ace_Map_Index_Table

Description: This table converts a protocol index into an ACE map index.

Table 73. Acl_Protocol_Ace_Map_Index_Table Register Parameters

Parameter	Value
Base Address	0x0004_2800
Register Size	2048
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	512
Record Spacing	4

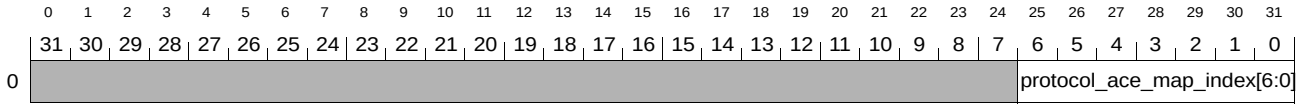


Figure 59. Acl_Protocol_Ace_Map_Index_Table Register Diagram

Table 74. Acl_Protocol_Ace_Map_Index_Table Field Parameters

Field Name	Parameters	Description
protocol_ace_map_index[6:0]	Mode = R/W Offset = 0.24	The ACE map index value.

This table is addressed by the concatenation of the `acl_index[5:0]` value and a protocol index value `(acl_protocol_index[2:0])`. Each entry in this table is a 7-bit index into `Acl_Protocol_Port_Ace_Map_Table`.

This table allows the eight protocol indexes appearing in 48 ACLs to address as many as 256 ACE maps. The following figure shows where this table fits in the processing pipeline.

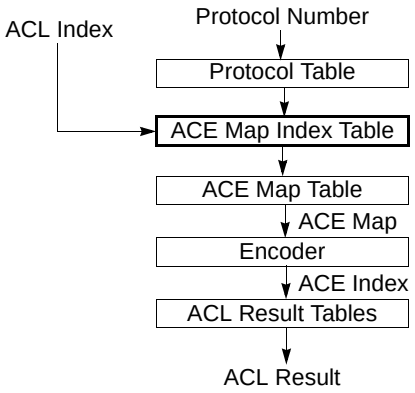


Figure 60. ACL Processing Pipeline

Appendix A: Registers (continued)

Acl_Protocol_Ace_Map_Table

Description: This table converts ACE map index values derived from protocol indexes into ACE maps.

Table 75. Acl_Protocol_Ace_Map_Table Register Parameters

Parameter	Value
Base Address	0x0004_1400
Register Size	1024
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	128
Record Spacing	8

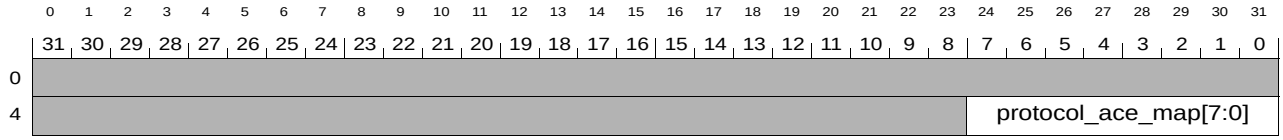


Figure 61. Acl_Protocol_Ace_Map_Table Register Diagram

Table 76. Acl_Protocol_Ace_Map_Table Field Parameters

Field Name	Parameters	Description
protocol_ace_map[7:0]	Mode = R/W Offset = 4.24	The ACE map value.

This table is addressed by `Acl_Protocol_Ace_Map_Index_Table.protocol_ace_map_index[6:0]` and returns the 8-bit ACE map value for the associated packet protocol. An ACE map is a vector that identifies all of the ACEs for which the associated protocol registers a match.

The following figure shows where this table fits in the ACL processing pipeline.

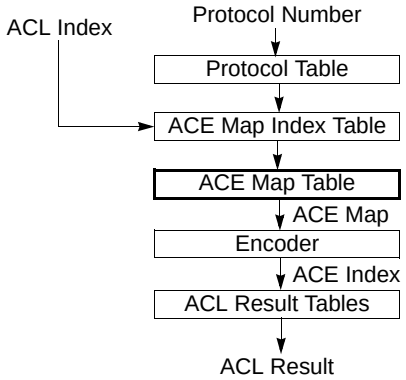


Figure 62. ACL Processing Pipeline

Appendix A: Registers (continued)

Acl_Protocol_Table

Description: This table encodes a packet's Ethertype and IP protocol indexes into a compact protocol index.

Table 77. Acl_Protocol_Table Register Parameters

Parameter	Value
Base Address	0x0004_2200
Register Size	256
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	64
Record Spacing	4

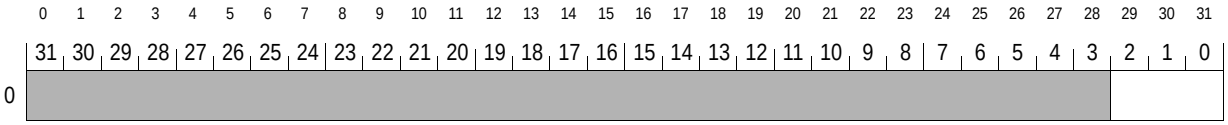


Figure 63. Acl_Protocol_Table Register Diagram

Table 78. Acl_Protocol_Table Field Parameters

Field Name	Parameters	Description
acl_protocol[2:0]	Mode = R/W Offset = 0.29 Instances = 1	An index value that is representative of the packet's Ethertype and IP protocol values.

This table is addressed by a concatenation of `ethertype_index[2:0]` (derived from the packet's Layer 2 type field) and `ip_protocol_index[2:0]` (derived from the packet's Layer 3 protocol field). `ethertype_index[2:0]` makes up the most significant portion of the table's address. This 6-bit concatenation is shifted left two bits in order to address 32-bit words.

Table 79. ethertype_index[2:0] and ip_protocol_index[2:0] Defined

ethertype_index[2:0]	Ethertype	protocol_index[2:0]	Protocol
0	IPv4	0	ICMP
1	IPv6	1	IGMP
2	ARP	2	TCP
3	RARP	3	UDP
4	user_type_0	4	user_protocol_0
5	user_type_1	5	user_protocol_1
6	user_type_2	6	user_protocol_2
7	unknown	7	unknown

Appendix A: Registers (continued)

Acl_Result_Table

Description: This register returns a priority adjustment command based on the ACL number and ACE number.

Table 80. Acl_Result_Table Register Parameters

Parameter	Value
Base Address	0x0003_4000
Register Size	16384
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	384
Record Spacing	4

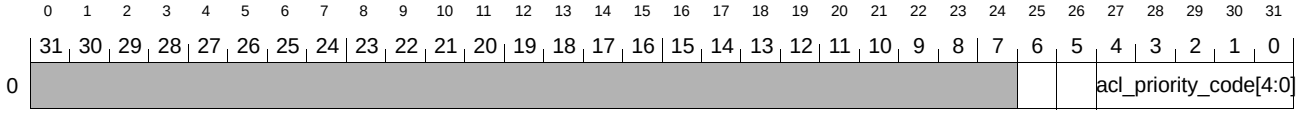


Figure 64. Acl_Result_Table Register Diagram

Table 81. Acl_Result_Table Field Parameters

Field Name	Parameters	Description
acl_permit	Mode = R/W Offset = 0.25 Instances = 1	When asserted, the packet is permitted access to the switching system. Otherwise, the denial mask is applied to the packet's destination map.
acl_log	Mode = R/W Offset = 0.26 Instances = 1	When asserted, the supervisor's ACL logging port is added to the packet's destination map. Logging and packet denial are independent.
acl_priority_code[4:0]	Mode = R/W Offset = 0.27 Instances = 1	The ACL priority opcode and priority value.

`acl_priority_code[4]` defines the action taken by the ACL function. These functions are defined below:

0 = preserve packet's priority value
1 = replace packet's priority value with `acl_priority_code[3:0]`

This table is addressed by a concatenation of `acl_index[5:0]` (that identifies a particular ACL) and `ace_index[2:0]` (that identifies a particular ACE). `acl_index[5:0]` occupies the upper portion of the concatenated address word. Three bits of zero are placed between `acl_index[5:0]` and `ace_index[2:0]` such that each increment of `acl_index[5:0]` advances the pointer into `Acl_Result_Table` by 64 entries.

`ace_index[2:0]` is an internal value (not accessible to the supervisor) that is derived from the priority encoding of the bitwise **and** of all of the ACE maps retrieved for a particular packet, hence, identifying the matching ACE.

Appendix A: Registers (continued)

Acl_Tcp_Key_Table_0

Description: This table is reserved and must be programmed to all ones for proper ACL operation.

Table 82. Acl_Tcp_Key_Table_0 Register Parameters

Parameter	Value
Base Address	0x0004_24f0
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

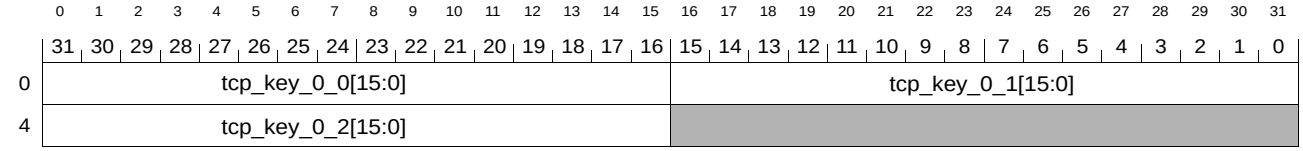


Figure 65. Acl_Tcp_Key_Table_0 Register Diagram

Table 83. Acl_Tcp_Key_Table_0 Field Parameters

Field Name	Parameters	Description
reserved_0_{0..2}[15:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 2.0	All fields must be filled with all ones for proper operation; (i.e., program tcp_key_0_0[15:0] to 0xFFFF, tcp_key_0_1[15:0] to 0xFFFF, and tcp_key_0_2[15:0] to 0xFFFF.)

This register is reserved and is a placeholder for future expansion. All fields must be filled with all ones for proper operation.

Appendix A: Registers (continued)

Acl_Tcp_Key_Table_1

Description: This table is reserved and must be programmed to all ones for proper ACL operation.

Table 84. Acl_Tcp_Key_Table_1 Register Parameters

Parameter	Value
Base Address	0x0004_24c0
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	8

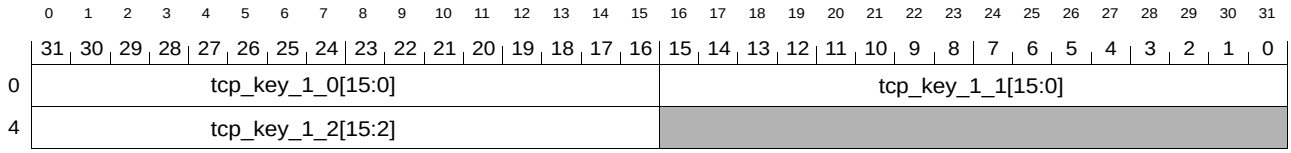


Figure 66. Acl_Tcp_Key_Table_1 Register Diagram

Table 85. Acl_Tcp_Key_Table_1 Field Parameters

Field Name	Parameters	Description
tcp_key_1_{0..2}[15:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 2.0	All fields must be filled with all ones for proper operation; (i.e., program tcp_key_1_0[15:0] to 0xFFFF, tcp_key_1_1[15:0] to 0xFFFF, and tcp_key_1_2[15:0] to 0xFFFF.)

This register is reserved and is a placeholder for future expansion. All fields must be filled with all ones for proper operation.

Appendix A: Registers (continued)

Acl_Tcp_Key_Table_2

Description: This table performs the first stage of TCP port number look-up.

Table 86. Acl_Tcp_Key_Table_2 Register Parameters

Parameter	Value
Base Address	0x0004_3000
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	8

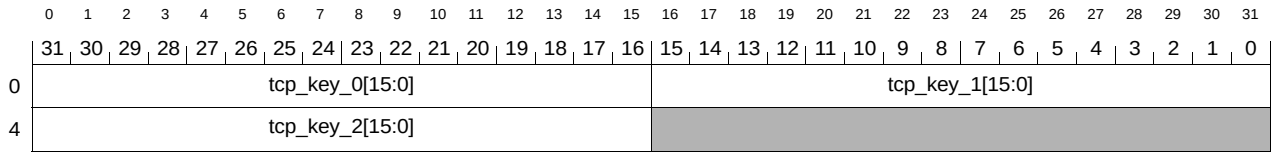


Figure 67. Acl_Tcp_Key_Table_2 Register Diagram

Table 87. Acl_Tcp_Key_Table_2 Field Parameters

Field Name	Parameters	Description
tcp_key_{0..2}[15:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 2.0	A set of 16-bit TCP port number values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This first stage of TCP port number look-up utilizes just a single record. Three TCP port number keys are stored here. The comparisons between the search argument and the keys result in a selection of one of four index values. These index values are used to address records in the next stage of the look-up.

Appendix A: Registers (continued)

Acl_Tcp_Key_Table_3

Description: This table performs the second stage of TCP port number look-up.

Table 88. Acl_Tcp_Key_Table_3 Register Parameters

Parameter	Value
Base Address	0x0004_2000
Register Size	32
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	4
Record Spacing	8

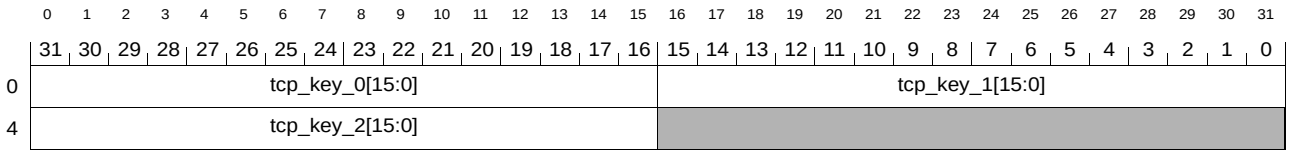


Figure 68. Acl_Tcp_Key_Table_3 Register Diagram

Table 89. Acl_Tcp_Key_Table_3 Field Parameters

Field Name	Parameters	Description
tcp_key_{0..2}[15:0]	Mode = R/W Offset = 0.0 Instances = 3 Spacing = 2.0	A set of 16-bit TCP port number values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This second stage of TCP port number look-up utilizes four records. Three TCP port number keys are stored in each record. The comparisons between the search argument and the keys result in a selection of one of four index values. These index values are used to address records in the next stage of the look-up.

Appendix A: Registers (continued)

Acl_Tcp_Key_Table_4

Description: This table performs the third and final stage of TCP port number look-up.

Table 90. Acl_Tcp_Key_Table_4 Register Parameters

Parameter	Value
Base Address	0x0003_f000
Register Size	256
Register Instances	1
Register Spacing	NA
Record Size	12
Record Instances	16
Record Spacing	16

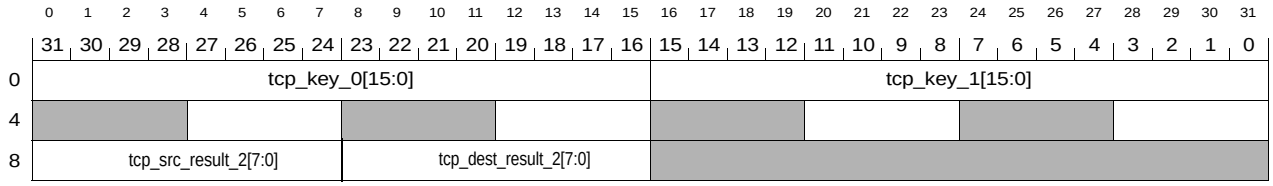


Figure 69. Acl_Tcp_Key_Table_4 Register Diagram

Table 91. Acl_Tcp_Key_Table_4 Field Parameters

Field Name	Parameters	Description
tcp_key_{0}[15:0]	Mode = R/W Offset = 0.0 Instances = 1	A 16-bit TCP port number value. This value is compared against the search argument. The result of this comparison is used to select one of two result values.
tcp_key_{1}[15:0]	Mode = R/W Offset = 0.16 Instances = 1	All fields must be filled with all ones for proper operation; (i.e. program tcp_key_1[15:0] to 0xFFFF.)
tcp_src_result_{0..1}[3:0]	Mode = R/W Offset = 4.4 Instances = 2 Spacing = 2.0	These 8-bit result values are returned as the final output from the look-up process. These results correspond to source port look-ups.
tcp_dest_result_{0..1}[3:0]	Mode = R/W Offset = 4.12 Instances = 2 Spacing = 2.0	These 8-bit result values are returned as the final output from the look-up process. These results correspond to destination port look-ups.
tcp_src_result_{2}[7:0]	Mode = R/W Offset = 8.0 Instances = 1	This field must be filled with all ones for proper operation; (i.e. tcp_src_result_{2}[7:0] to 0xFF.)
tcp_dest_result_{2}[7:0]	Mode = R/W Offset = 8.8 Instances = 1	This field must be filled with all ones for proper operation; (i.e. tcp_dest_result_{2}[7:0] to 0xFF.)

This third and final stage of TCP port number look-up utilizes 16 records. One TCP port number key is used in each record. The comparisons between the search argument and the keys result in a selection of one of two result values. These are returned as the final output of the look-up process.

Appendix A: Registers (continued)

Acl_Vlan_Index_Table

Description: This table is used to map the 8-bit VLAN index to one of 64 ACL numbers.

Table 92. Acl_Vlan_Index_Table Register Parameters

Parameter	Value
Base Address	0x0004_1000
Register Size	1024
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	256
Record Spacing	4

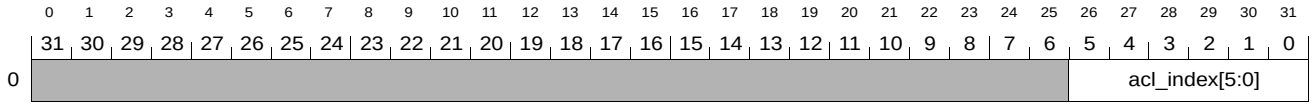


Figure 70. Acl_Vlan_Index_Table Register Diagram

Table 93. Acl_Vlan_Index_Table Field Parameters

Field Name	Parameters	Description
acl_index[5:0]	Mode = R/W Offset = 0.26 Instances = 1	The ACL index number.

The 8-bit VLAN index assigned to the receive packet is used as an address into this table. The addressed value is used as the ACL index for the packet. Essentially, this table provides a 256-to-48 mapping function. The largest value allowed for acl_index[5:0] is 47.

Appendix A: Registers (continued)

Device_Version

Description: This register returns version number for the device.

Table 94. Device_Version Register Parameters

Parameter	Value
Base Address	0x000c_8320
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

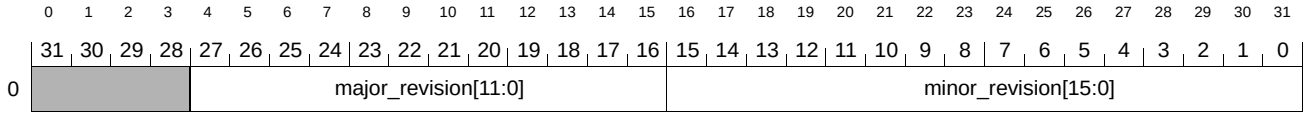


Figure 71. Device_Version Register Diagram

Table 95. Device_Version Field Parameters

Field Name	Parameter	Description
major_revisionision[11:0]	Mode = RO Offset = 0.4 Instances = 1	The device's major revision number. The major revision number reflects the iteration from one all-layer spin to the next. This field returns a value in a binary-coded decimal form (each decimal digit occupies a 4-bit space). In binary-coded format, the value of this field for the various versions is: Revision B: 002 Revision B1: 002 Revision C: 003
minor_revision[15:0]	Mode = RO Offset = 0.16 Instances = 1	The device's minor revision number. The minor revision number reflects the iteration from one metal layer-only spin to the next. This field returns a value in a binary-coded decimal form (each decimal digit occupies a 4-bit space). In binary-coded format, the value of this field for the various versions is: Revision B: 0000 Revision B1: 0001 Revision C: 0000

Appendix A: Registers (continued)

Layer_2_Active_Port_Map

Description: Defines which ports are enabled for the normal forwarding of traffic.

Table 96. Layer_2_Active_Port_Map Register Parameters

Parameter	Value
Base Address	0x000c_4620
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

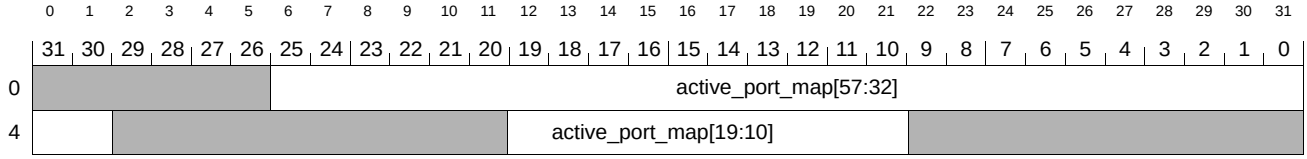


Figure 72. Layer_2_Active_Port_Map Register Diagram

Table 97. Layer_2_Active_Port_Map Field Parameters

Field Name	Parameter	Description
active_port_map[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The active port map value. Each bit in the map corresponds to an Ethernet port or one of the supervisor's queues.

This field is used to identify those ports that are enabled to forward Ethernet traffic in a normal manner. This information is used in determining whether or not a packet has been prevented from being forwarded due to VLAN mismatches.

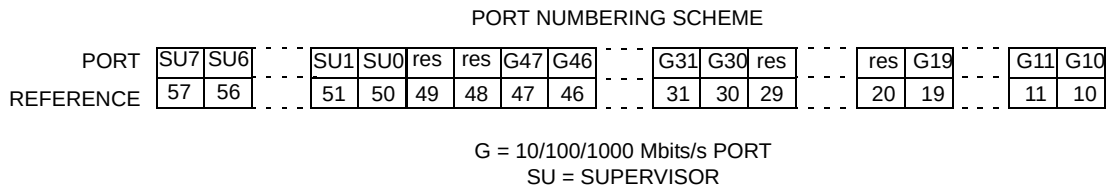


Figure 73. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Aggregation_Mask_Table

Description: Enables the selection of a port from within an aggregate of ports.

Table 98. Layer_2_Aggregation_Mask_Table Register Parameters

Parameter	Value
Base Address	0x000c_4480
Register Size	64
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	8
Record Spacing	8

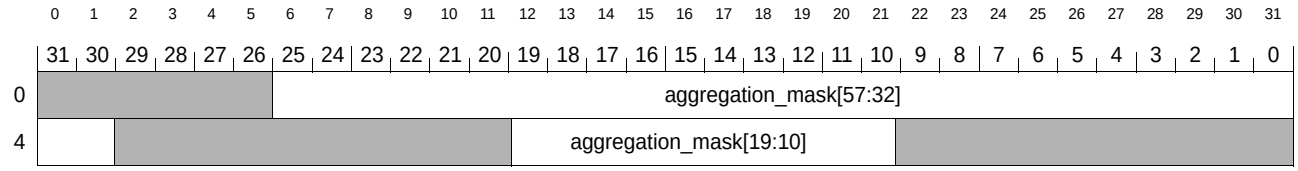


Figure 74. Layer_2_Aggregation_Mask_Table Register Diagram

Table 99. Layer_2_Aggregation_Mask_Table Field Parameters

Field Name	Parameter	Description
aggregation_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The aggregation mask value. Each bit in the mask corresponds to an Ethernet port or one of the supervisor's queues.

Through a series of addition reductions, the receive packet's MAC destination and MAC source addresses are reduced to a 3-bit index. Any particular combination of MAC address values always reduces to the same 3-bit index. This index is used to select one of eight mask values from this table.

When the look-up on a packet's destination address indicates that its destination port is part of an aggregation of ports, then all of the ports in that aggregate are enabled by the initial destination map. The mask value retrieved from this table is used to select one of the ports from within the aggregate.

This set of eight aggregate masks is shared among all of the aggregates configured in the system.

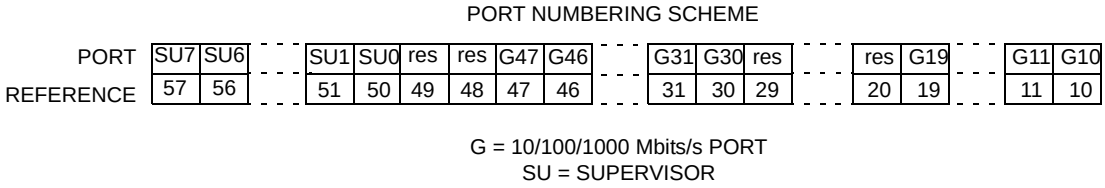


Figure 75. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Blocking_Mask

Description: Disables those ports that should not receive packets when a receive port is in blocking mode.

Table 100. Layer_2_Blocking_Mask Register Parameters

Parameter	Value
Base Address	0x000c_4628
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

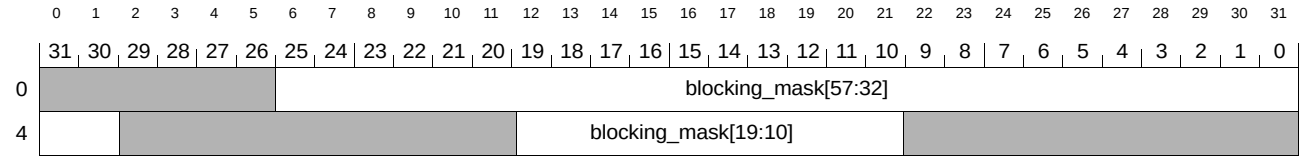


Figure 76. Layer_2_Blocking_Mask Register Diagram

Table 101. Layer_2_Blocking_Mask Field Parameters

Field Name	Parameter	Description
blocking_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The blocking mask value. Each bit in the mask corresponds to an Ethernet port or one of the supervisor's queues.

This mask is applied to the destination port map if the packet's source port is in the blocking state according to spanning tree rules. Bits asserted in `blocking_mask[57:10]` have the effect of disabling the corresponding destinations.

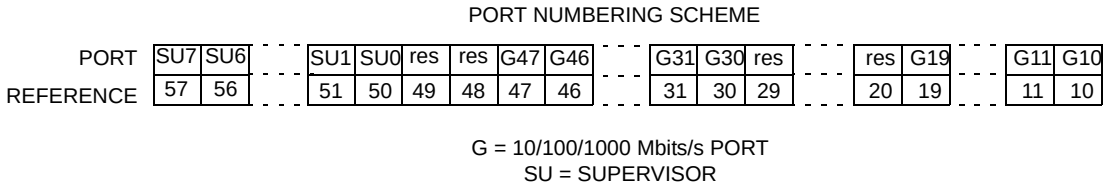


Figure 77. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Current_Time

Description: The time value used for MAC source address time stamp updates.

Table 102. Layer_2_Current_Time Register Parameters

Parameter	Value
Base Address	0x000c_4660
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

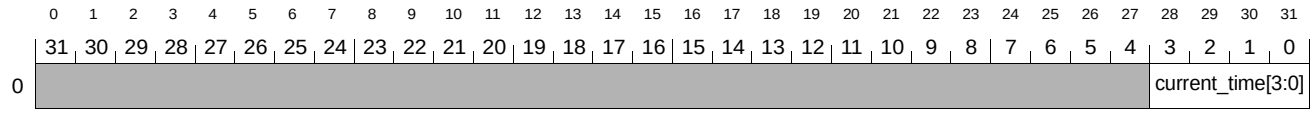


Figure 78. Layer_2_Current_Time Register Diagram

Table 103. Layer_2_Current_Time Field Parameters

Field Name	Parameter	Description
current_time[3:0]	Mode = R/W Offset = 0.28 Instances = 1	The current time stamp setting.

When a known source address is received, its time stamp is updated in `Layer_2_Time_Stamp_Table`. The value written to that table is `Layer_2_Current_Time.current_time[3:0]`.

Appendix A: Registers (continued)

Layer_2_Dest_Map_Table

Description: The table of initial destination port maps.

Table 104. Layer_2_Dest_Map_Table Register Parameters

Parameter	Value
Base Address	0x000c_2000
Register Size	2048
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	256
Record Spacing	8

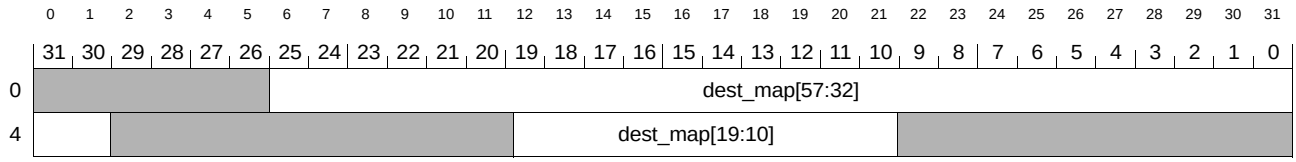


Figure 79. Layer_2_Dest_Map_Table Register Diagram

Table 105. Layer_2_Dest_Map_Table Field Parameters

Field Name	Parameter	Description
dest_map[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The initial destination port map.

This table is addressed by the `layer_2_dest_map_index[7:0]` value returned as part of the associated data from the destination MAC address look-up. The value retrieved from this table is a destination port map. Bits are asserted in these map values to indicate destinations for the packet. This initial port map is adjusted through several steps of masking (eliminating destinations) and mapping (adding destinations).

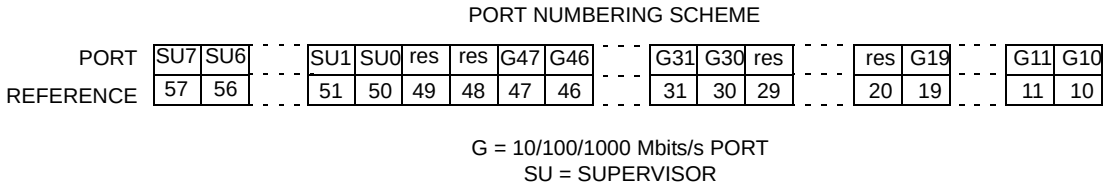


Figure 80. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Dest_Mirror_Map

Description: A map of the destination ports configured to be mirrored.

Table 106. Layer_2_Dest_Mirror_Map Register Parameters

Parameter	Value
Base Address	0x000c_4630
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

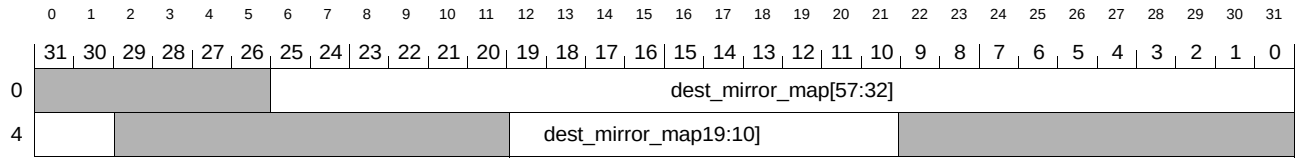


Figure 81. Layer_2_Dest_Mirror_Map Register Diagram

Table 107. Layer_2_Dest_Mirror_Map Field Parameters

Field Name	Parameter	Description
dest_mirror_map[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The destination mirroring port map.

If a packet is being forwarded to a port that corresponds to a bit asserted in `dest_mirror_map[57:10]`, then that packet is copied to the system's mirror port.

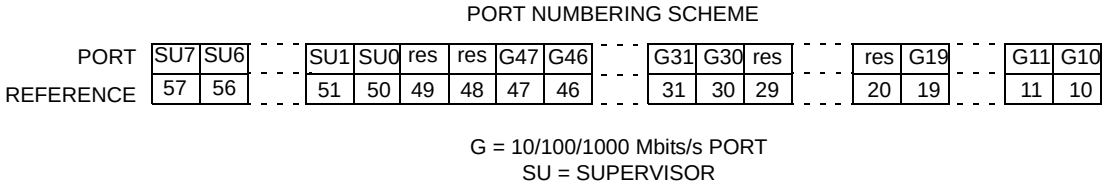


Figure 82. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Flood_Map

Description: Identifies the ports to which unknown destination packets are flooded.

Table 108. Layer_2_Flood_Map Register Parameters

Parameter	Value
Base Address	0x000c_4638
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

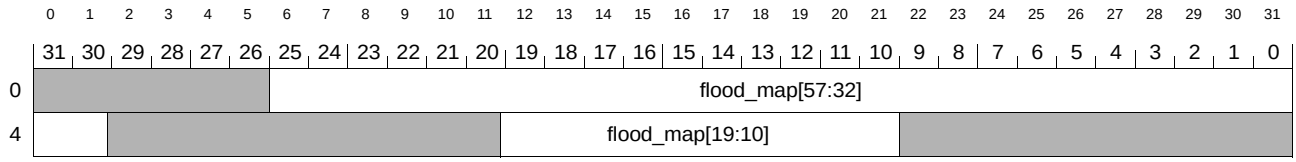


Figure 83. Layer_2_Flood_Map Register Diagram

Table 109. Layer_2_Flood_Map Field Parameters

Field Name	Parameter	Description
flood_map[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The flood map.

When a receive packet's destination address is not found in the Layer 2 address table, this flood map is used as the initial destination map.

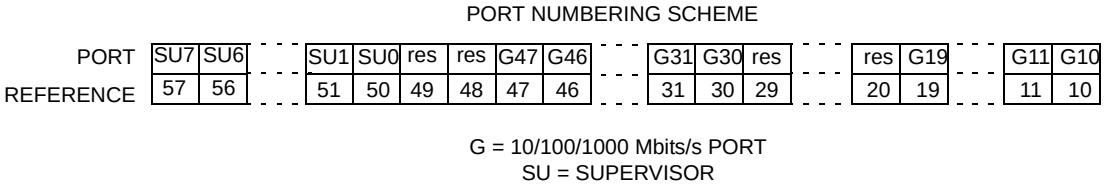


Figure 84. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Global_Mask

Description: Disables destination ports.

Table 110. Layer_2_Global_Mask Register Parameters

Parameter	Value
Base Address	0x000c_4640
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

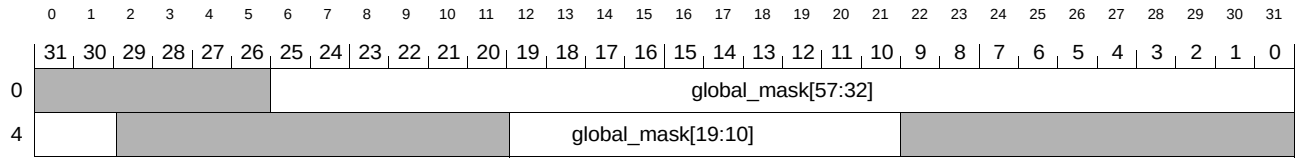


Figure 85. Layer_2_Global_Mask Register Diagram

Table 111. Layer_2_Global_Mask Field Parameters

Field Name	Parameter	Description
global_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The global destination mask.

Setting a bit in this register disables packets from being forwarded to the corresponding port. Supervisor transmissions are not affected by `Layer_2_Global_Mask`.

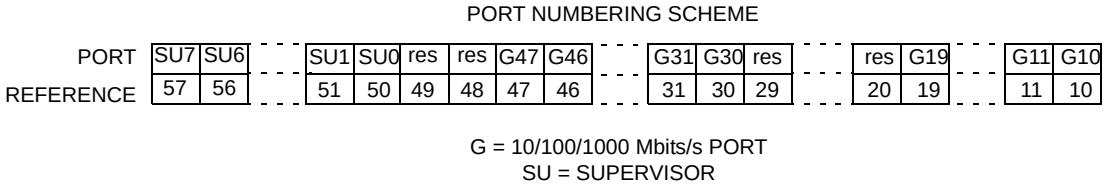


Figure 86. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Igmp_Snooping_Port

Description: Identifies the port to be used for IGMP snooping.

Table 112. Layer_2_Igmp_Snooping_Port Register Parameters

Parameter	Value
Base Address	0x000c_4664
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

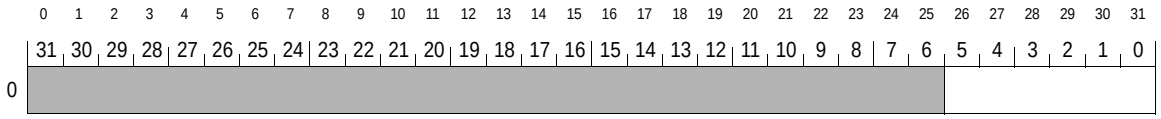


Figure 87. Layer_2_Igmp_Snooping_Port Register Diagram

Table 113. Layer_2_Igmp_Snooping_Port Field Parameters

Field Name	Parameter	Description
igmp_snooping_port[5:0]	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	The port number used for IGMP snooping.

When so enabled, IGMP packets are copied to the port number specified here.

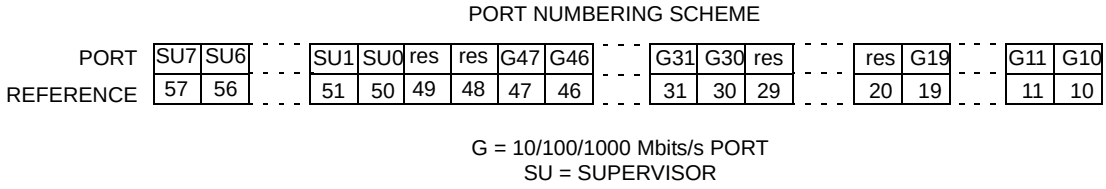


Figure 88. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Key_Table_0

Description: The first stage of a MAC address look-up is performed by this table.

Table 114. Layer_2_Key_Table_0 Register Parameters

Parameter	Value
Base Address	0x000c_4600
Register Size	24
Register Instances	1
Register Spacing	NA
Record Size	24
Record Instances	1
Record Spacing	NA

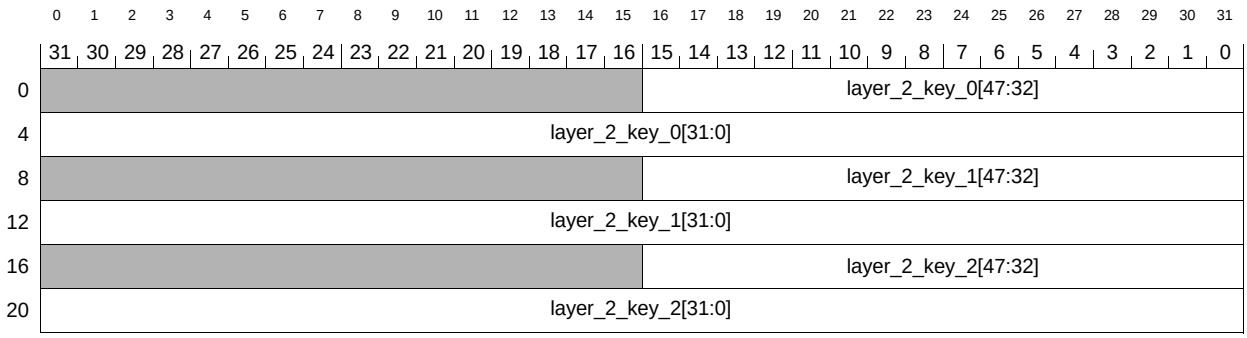


Figure 89. Layer_2_Key_Table_0 Register Diagram

Table 115. Layer_2_Key_Table_0 Field Parameters

Field Name	Parameter	Description
layer_2_key_{0..2}[47:0]	Mode = R/W Offset = 0.16 Instances = 3 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This first stage of a MAC address look-up involves the use of a single record containing three keys. The three keys are compared against the search argument. The results of these comparisons serve to select one of the four index values for use in the next stage of the look-up.

Appendix A: Registers (continued)

Layer_2_Key_Table_1

Description: The second stage of a MAC address look-up is performed by this table.

Table 116. Layer_2_Key_Table_1 Register Parameters

Parameter	Value
Base Address	0x000c_4400
Register Size	128
Register Instances	1
Register Spacing	NA
Record Size	24
Record Instances	4
Record Spacing	32

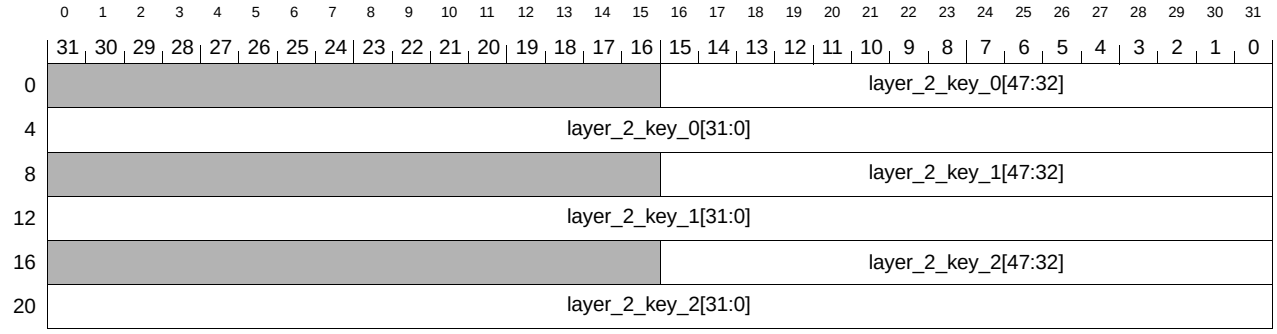


Figure 90. Layer_2_Key_Table_1 Register Diagram

Table 117. Layer_2_Key_Table_1 Field Parameters

Field Name	Parameter	Description
layer_2_key_{}{0..2}[47:0]	Mode = R/W Offset = 0.16 Instances = 3 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This second stage of a MAC address look-up involves the use of four records, each containing three keys. The three keys are compared against the search argument. The results of these comparisons serve to select one of the four index values for use in the next stage of the look-up.

Appendix A: Registers (continued)

Layer_2_Key_Table_2

Description: The third stage of a MAC address look-up is performed by this table.

Table 118. Layer_2_Key_Table_2 Register Parameters

Parameter	Value
Base Address	0x000c_4000
Register Size	512
Register Instances	1
Register Spacing	NA
Record Size	24
Record Instances	16
Record Spacing	32

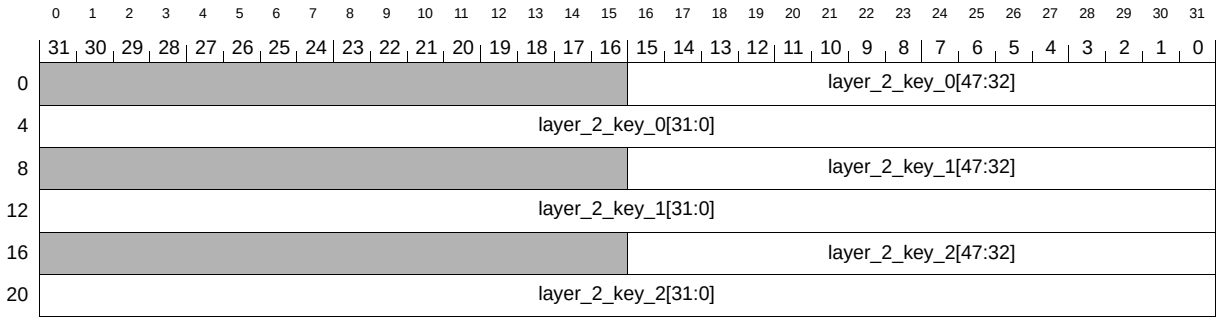


Figure 91. Layer_2_Key_Table_2 Register Diagram

Table 119. Layer_2_Key_Table_2 Field Parameters

Field Name	Parameter	Description
layer_2_key_{0..2}[47:0]	Mode = R/W Offset = 0.16 Instances = 3 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This third stage of a MAC address look-up involves the use of 16 records, each containing three keys. The three keys are compared against the search argument. The results of these comparisons serve to select one of the four index values for use in the next stage of the look-up.

Appendix A: Registers (continued)

Layer_2_Key_Table_3

Description: The fourth stage of a MAC address look-up is performed by this table.

Table 120. Layer_2_Key_Table_3 Register Parameters

Parameter	Value
Base Address	0x000c_3000
Register Size	2048
Register Instances	1
Register Spacing	NA
Record Size	32
Record Instances	64
Record Spacing	32

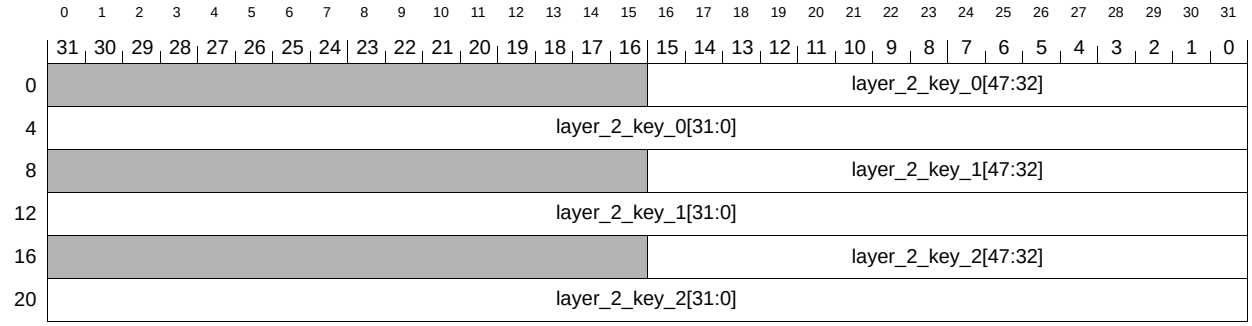


Figure 92. Layer_2_Key_Table_3 Register Diagram

Table 121. Layer_2_Key_Table_3 Field Parameters

Field Name	Parameter	Description
layer_2_key_{0..2}[47:0]	Mode = R/W Offset = 0.16 Instances = 3 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This fourth stage of a MAC address look-up involves the use of 64 records, each containing three keys. The three keys are compared against the search argument. The results of these comparisons serve to select one of the four index values for use in the next stage of the look-up.

Appendix A: Registers (continued)

Layer_2_Key_Table_4

Description: The fifth stage of a MAC address look-up is performed by this table.

Table 122. Layer_2_Key_Table_4 Register Parameters

Parameter	Value
Base Address	0x000c_0000
Register Size	8192
Register Instances	1
Register Spacing	NA
Record Size	32
Record Instances	256
Record Spacing	32

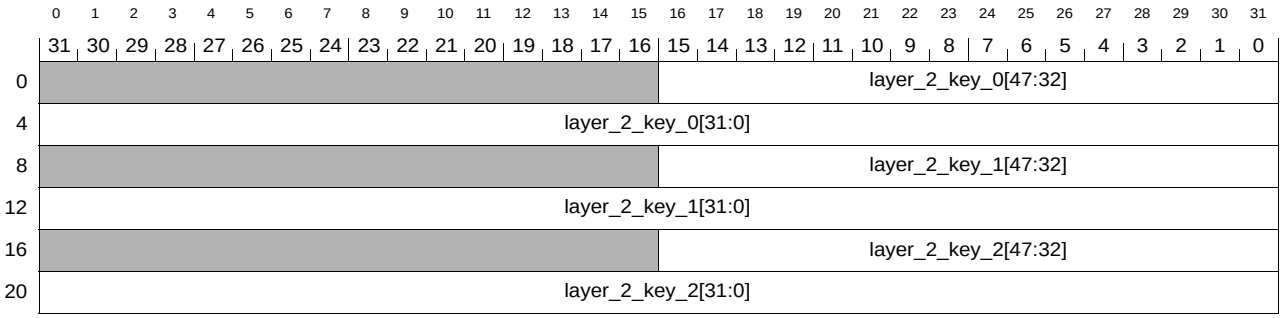


Figure 93. Layer_2_Key_Table_4 Register Diagram

Table 123. Layer_2_Key_Table_4 Field Parameters

Field Name	Parameter	Description
layer_2_key_{0..2}[47:0]	Mode = R/W Offset = 0.16 Instances = 3 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This fifth stage of a MAC address look-up involves the use of 256 records, each containing three keys. The three keys are compared against the search argument. The results of these comparisons serve to select one of the four index values for use in the next stage of the look-up.

Appendix A: Registers (continued)

Layer_2_Key_Table_5

Description: The sixth stage of a MAC address look-up is performed by this table.

Table 124. Layer_2_Key_Table_5 Register Parameters

Parameter	Value
Base Address	0x000b_0000
Register Size	32768
Register Instances	1
Register Spacing	NA
Record Size	32
Record Instances	1024
Record Spacing	32

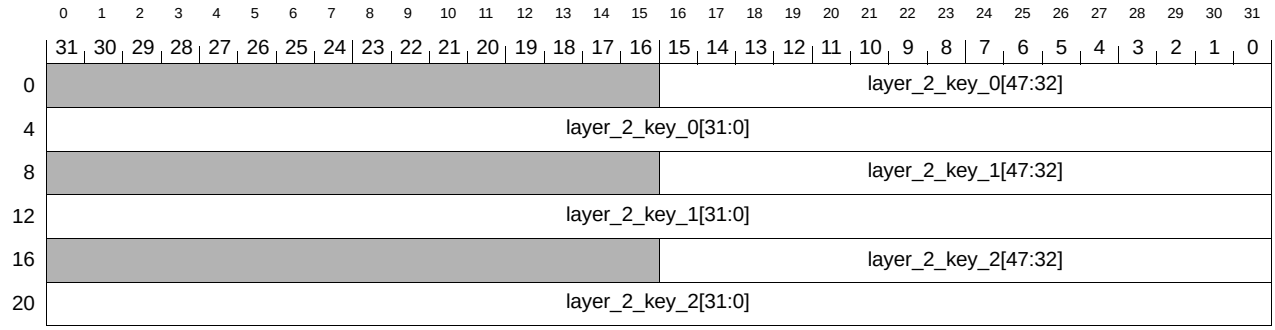


Figure 94. Layer_2_Key_Table_5 Register Diagram

Table 125. Layer_2_Key_Table_5 Field Parameters

Field Name	Parameter	Description
layer_2_key_{0..2}[47:0]	Mode = R/W Offset = 0.16 Instances = 3 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of four index values for use in the next stage of the look-up.

This sixth stage of a MAC address look-up involves the use of 1,024 records, each containing three keys. The three keys are compared against the search argument. The results of these comparisons serve to select one of the four index values for use in the next stage of the look-up.

Appendix A: Registers (continued)

Layer_2_Key_Table_6

Description: The seventh and final stage of a MAC address look-up is performed by this table.

Table 126. Layer_2_Key_Table_6 Register Parameters

Parameter	Value
Base Address	0x0008_0000
Register Size	131072
Register Instances	1
Register Spacing	NA
Record Size	24
Record Instances	4096
Record Spacing	32

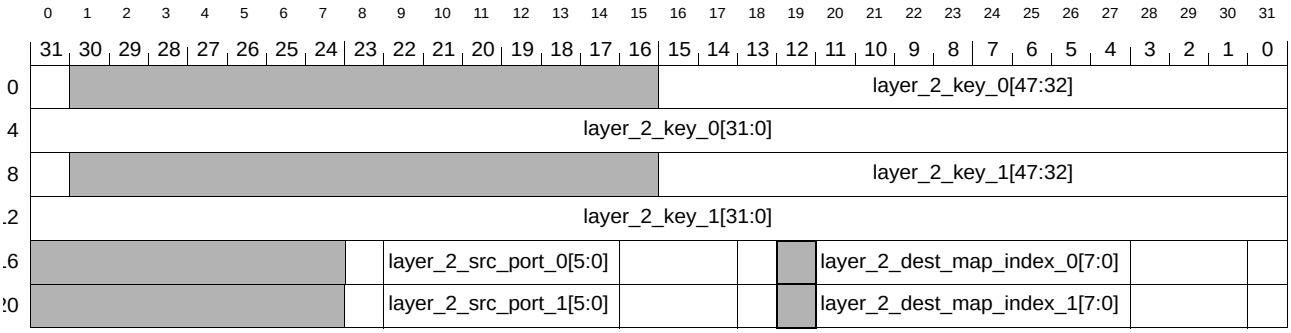


Figure 95. Layer_2_Key_Table_6 Register Diagram

Table 127. Layer_2_Key_Table_6 Field Parameters

Field Name	Parameters	Description
key_valid_{0..1}	Mode = R/W Offset = 0.0 Instances = 2 Spacing = 8.0	This bit is asserted to indicate that the corresponding key table entry is valid. If an invalid key is found at the end of a look-up, then a look-up miss indication is returned in the same manner as if a valid entry had not matched the search argument.
layer_2_key_{0..1}[47:0]	Mode = R/W Offset = 0.16 Instances = 2 Spacing = 8.0	A set of 48-bit MAC address values. These values are compared against the search argument. The results of these comparisons are used to select one of two associated data values.
layer_2_src_permit_{0..1}	Mode = R/W Offset = 16.8 Instances = 2 Spacing = 4.0	This bit must be asserted for a packet whose source address matches a valid associated key to be forwarded.

Appendix A: Registers (continued)

Layer_2_Key_Table_6 (continued)

Table 127. Layer_2_Key_Table_6 Field Parameters (continued)

Field Name	Parameters	Description
layer_2_src_port_{0..1}[5:0]	Mode = R/W Offset = 16.9 Instances = 2 Spacing = 4.0	This field identifies the port via which the associated address was learned. If the logical receive port number of a packet with a matching source address does not match this field, then the attached source device has moved from one port to another and the address table must be updated.
layer_2_src_flow_id_{0..1}[2:0]	Mode = R/W Offset = 16.15 Instances = 2 Spacing = 4.0	This field forms part of a flow identifier. This value is used for policing purposes.
layer_2_src_log_{0..1}	Mode = R/W Offset = 16.18 Instances = 2 Spacing = 4.0	If this bit is set for a packet whose source address matches the associated key, then the packet is copied to the supervisor's logging queue.
layer_2_dest_map_index_{0..1}[7:0]	Mode = R/W Offset = 16.20 Instances = 2 Spacing = 4.0	This field is used to select one of the shared destination port map values.
layer_2_dest_flow_id_{0..1}[2:0]	Mode = R/W Offset = 16.28 Instances = 2 Spacing = 4.0	A flow identifier based on a packet's destination address. This value is used for policing purposes.
layer_2_dest_log_{0..1}	Mode = R/W Offset = 16.31 Instances = 2 Spacing = 4.0	If this bit is set for a packet whose destination address matches the associated key, then the packet is copied to the supervisor's logging queue.

This seventh and final stage of a MAC address look-up involves the use of 4,096 records, each containing two keys and two associated data values. The two keys are compared against the search argument. The results of these comparisons serve to select one of the two associated data values as the final output of the look-up.

Appendix A: Registers (continued)

Layer_2_Learning_Mask

Description: Disables those ports that should not transmit packets received via a port in the learning mode.

Table 128. Layer_2_Learning_Mask Register Parameters

Parameter	Value
Base Address	0x000c_4648
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

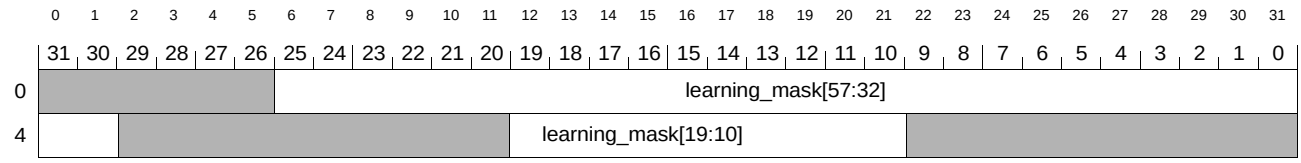


Figure 96. Layer_2_Learning_Mask Register Diagram

Table 129. Layer_2_Learning_Mask Field Parameters

Field Name	Parameter	Description
learning_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The learning mask value. Each bit in the mask corresponds to an Ethernet port or one of the supervisor's queues.

This mask is applied to the destination port map if the packet's source port is in the learning state according to spanning tree rules. Bits asserted in `learning_mask[57:10]` have the effect of disabling the corresponding destinations.

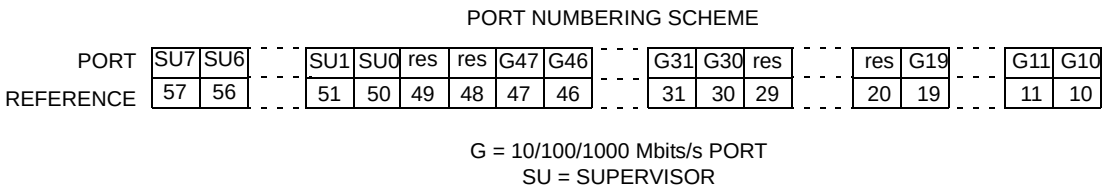


Figure 97. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Learning_Port

Description: Identifies the port to be used for MAC source address learning.

Table 130. Layer_2_Learning_Port Register Parameters

Parameter	Value
Base Address	0x000c_4668
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

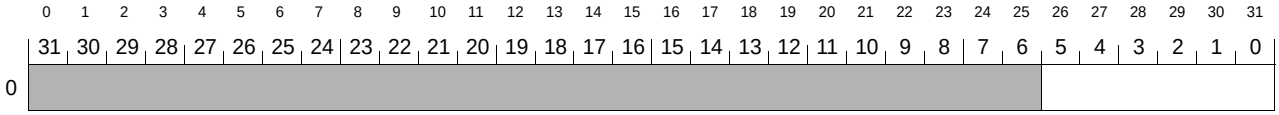


Figure 98. Layer_2_Learning_Port Register Diagram

Table 131. Layer_2_Learning_Port Field Parameters

Field Name	Parameter	Description
learning_port[5:0]	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	The port number used for MAC source address learning.

When a packet is received whose MAC source address is not found in the address table or whose source port listed in the address table does not match the logical port through which it was received, it is copied to the port number specified here.

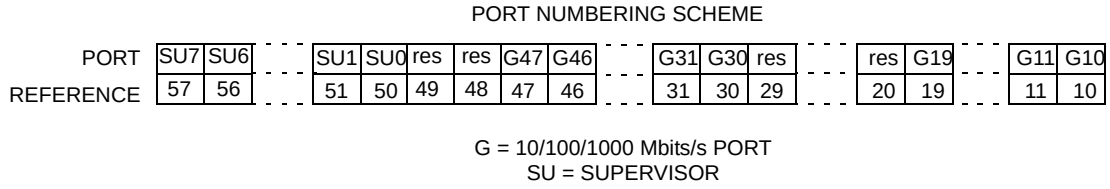


Figure 99. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Logical_Port_Table

Description: Translates physical port numbers to logical port numbers.

Table 132. Layer_2_Logical_Port_Table Register Parameters

Parameter	Value
Base Address	0x000c_4500
Register Size	256
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	50
Record Spacing	4

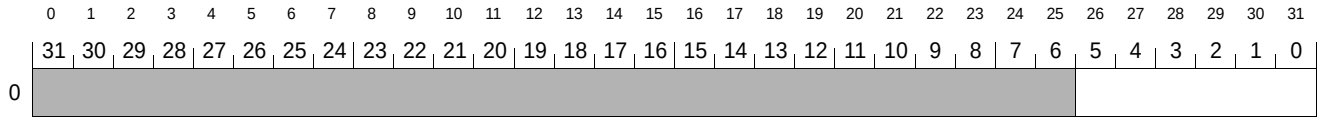


Figure 100. Layer_2_Logical_Port_Table Register Diagram

Table 133. Layer_2_Logical_Port_Table Field Parameters

Field Name	Parameter	Description
logical_port_{0..49}[5:0]	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	The logical port number to be used for a particular physical port.

The physical receive port number of a packet is used as an index into this table to determine the packet's logical receive port. Logical port numbers are used to allow multiple physical ports to share a single identity. This capability reduces learning thrashing when packets are received on ports that are members of an aggregate.

PORT NUMBERING SCHEME (TABLE INDEXING)

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 101. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Mirror_Port

Description: Identifies the system's mirror port.

Table 134. Layer_2_Mirror_Port Register Parameters

Parameter	Value
Base Address	0x000c_466c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

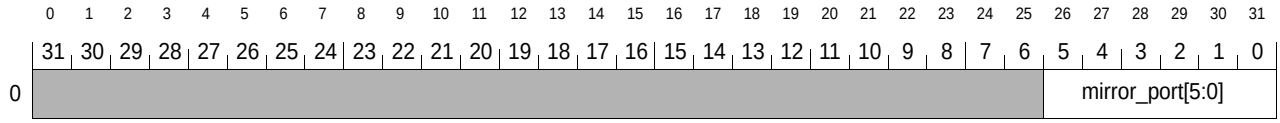


Figure 102. Layer_2_Mirror_Port Register Diagram

Table 135. Layer_2_Mirror_Port Field Parameters

Field Name	Parameter	Description
mirror_port[5:0]	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	The mirror port number.

When mirroring is enabled and a packet is received or transmitted via a port that is being mirrored, that packet is also copied to the port identified by this register.

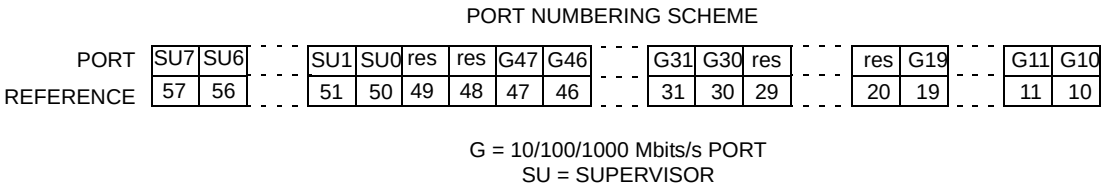


Figure 103. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Mode

Description: Assorted mode bits for Layer 2 processing.

Table 136. Layer_2_Mode Register Parameters

Parameter	Value
Base Address	0x000c_4700
Register Size	244
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	50
Record Spacing	4

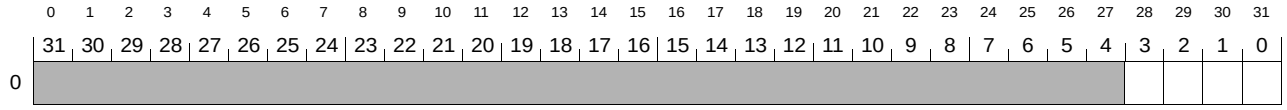


Figure 104. Layer_2_Mode Register Diagram

Table 137. Layer_2_Mode Field Parameters

Field Name	Parameters	Description
discard_unknown_src_en	Mode = R/W, Offset = 0.28, Instances = 1, Reset = 0	This bit enables the discarding of packet's whose source addresses are not present in the address table.
supervisor_route_en	Mode = R/W, Offset = 0.29, Instances = 1, Reset = 0	When this bit is asserted, packets whose destination addresses are known (even though there is a mismatch between the source's VLAN and the destination's) are forwarded to the supervisor for routing between the VLANs. Otherwise, such packets are not forwarded to normal transmit ports.
user_port_snoop_en	Mode = R/W, Offset = 0.30, Instances = 1, Reset = 0	Asserting this bit enables the snooping of a user-defined TCP destination port.
igmp_snoop_en	Mode = R/W, Offset = 0.31, Instances = 1, Reset = 0	Asserting this bit enables the copying of IGMP packets to the supervisor's IGMP snooping port. When deasserted, such packets are not copied to the supervisor.

This register contains an assortment of Layer 2 processing mode bits and fields. There is one record for each physical Ethernet port.

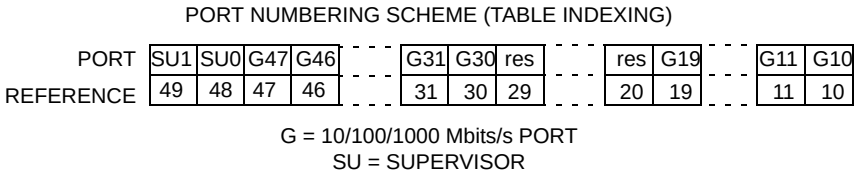


Figure 105. Port Numbering Scheme (Table Indexing)

Appendix A: Registers (continued)

Layer_2_No_Dest_Packets

Description: A count of the number of packets with a null destination map.

Table 138. Layer_2_No_Dest_Packets Register Parameters

Parameter	Value
Base Address	0x000c_4680
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

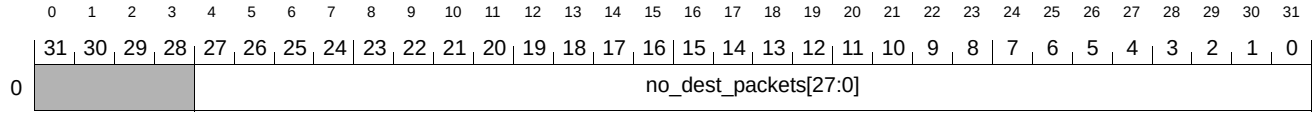


Figure 106. Layer_2_No_Dest_Packets Register Diagram

Table 139. Layer_2_No_Dest_Packets Field Parameters

Field Name	Parameter	Description
no_dest_packets[27:0]	Mode = R/W Offset = 0.4 Instances = 1 Reset = 0	The number of packets whose destination map in null.

If a packet's destination map is null (i.e., no destinations), then this counter is incremented.

Appendix A: Registers (continued)

Layer_2_Port_State_Table

Description: Maintains the spanning tree state on a per-port basis.

Table 140. Layer_2_Port_State_Table Register Parameters

Parameter	Value
Base Address	0x000a_0000
Register Size	62464
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	15616
Record Spacing	4

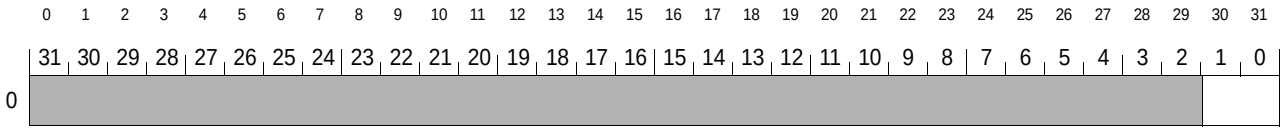


Figure 107. Layer_2_Port_State_Table Register Diagram

Table 141. Layer_2_Port_State_Table Field Parameters

Field Name	Parameter	Description
stp_state[1:0]	Mode = R/W Offset = 0.30 Instances = 1	The port STP state: 00_2 = blocking 01_2 = learning 10_2 = forwarding 11_2 = disabled

In order to support spanning tree, the spanning tree state of each port must be maintained. This table provides this information to the bridging function.

This table is addressed by the packet's receive port time 256:
`address[13:0] = rx_port[5:0] * 256`.

For proper operation, the appropriate spanning tree port state value must be written to 256 consecutive register word locations starting with the location derived as shown above.

PORT NUMBERING SCHEME

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 108. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Src_Deny_Mask

Description: Identifies those ports to be eliminated as destinations based on a source address denial.

Table 142. Layer_2_Src_Deny_Mask Register Parameters

Parameter	Value
Base Address	0x000c_4650
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

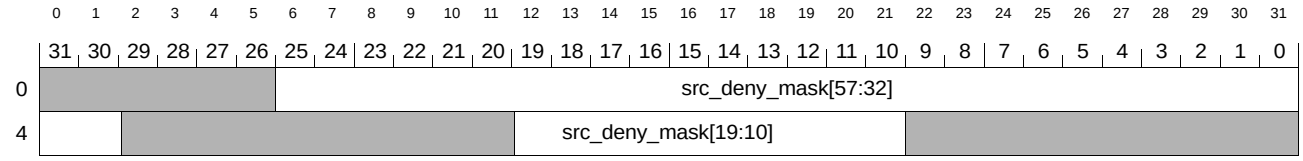


Figure 109. Layer_2_Src_Deny_Mask Register Diagram

Table 143. Layer_2_Src_Deny_Mask Field Parameters

Field Name	Parameter	Description
src_deny_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The source denial mask.

If the MAC source address look-up returns a source deny indication, then the ports identified by this mask are eliminated as destinations for the packet. Asserted bits in `src_deny_mask[57:10]` identify those ports that are eliminated (masked) by this function. Deasserted bits in `src_deny_mask[57:10]` have no effect on the packet's destination port map.

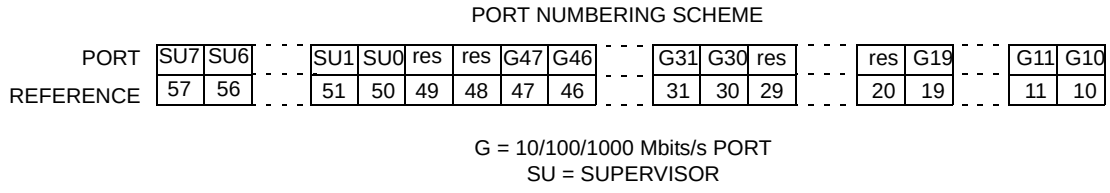


Figure 110. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Src_Mirror_Map

Description: Identifies those ports whose receive traffic is to be mirrored.

Table 144. Layer_2_Src_Mirror_Map Register Parameters

Parameter	Value
Base Address	0x000c_4658
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

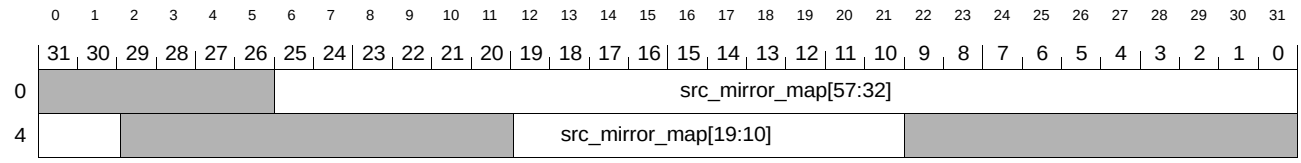


Figure 111. Layer_2_Src_Mirror_Map Register Diagram

Table 145. Layer_2_Src_Mirror_Map Field Parameters

Field Name	Parameter	Description
src_mirror_map[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The source mirror map.

This port map identifies those ports whose receive traffic is to be copied to the mirror port. If a packet's receive port's corresponding bit in `src_mirror_map[57:10]` is asserted, then the packet is copied to the mirror port.

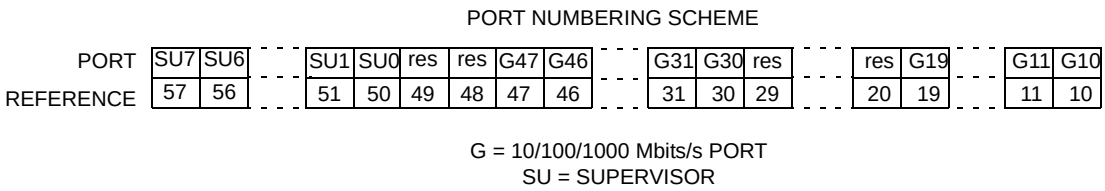


Figure 112. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Src_Port_Mask_Table

Description: Enables the limiting of destinations based on source port.

Table 146. Layer_2_Src_Port_Mask_Table Register Parameters

Parameter	Value
Base Address	0x000c_4200
Register Size	400
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	50
Record Spacing	8

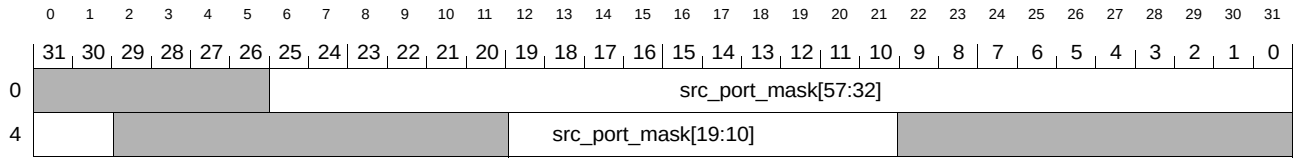


Figure 113. Layer_2_Src_Port_Mask_Table Register Diagram

Table 147. Layer_2_Src_Port_Mask_Table Field Parameters

Field Name	Parameter	Description
src_port_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The source port mask.

Source port masking is used to limit the destination ports reachable from each source port. There is a mask in the table for each of the system's source ports.

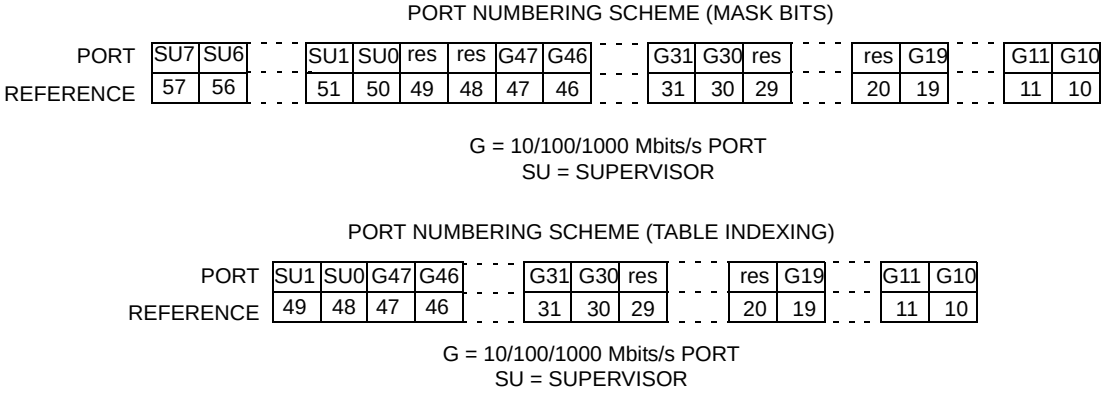


Figure 114. Port Numbering Scheme (Mask Bits and Table Indexing)

Appendix A: Registers (continued)

Layer_2_Supervisor_Route_Port

Description: Identifies the supervisor's route port.

Table 148. Layer_2_Supervisor_Route_Port Register Parameters

Parameter	Value
Base Address	0x000c_4674
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

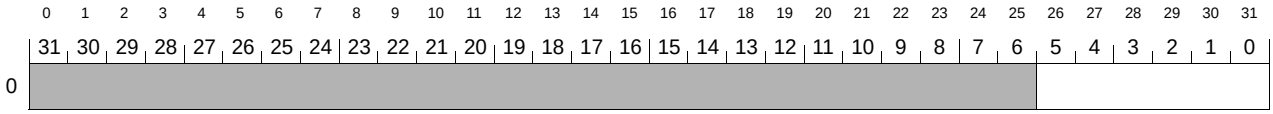


Figure 115. Layer_2_Supervisor_Route_Port Register Diagram

Table 149. Layer_2_Supervisor_Route_Port Field Parameters

Field Name	Parameter	Description
supervisor_route_port[5:0]	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	The supervisor route port number.

When a receive packet's MAC source address is found in the address table but the source VLAN and destination VLAN do not match and supervisor routing is enabled, this port is added to the packet's destination port map.

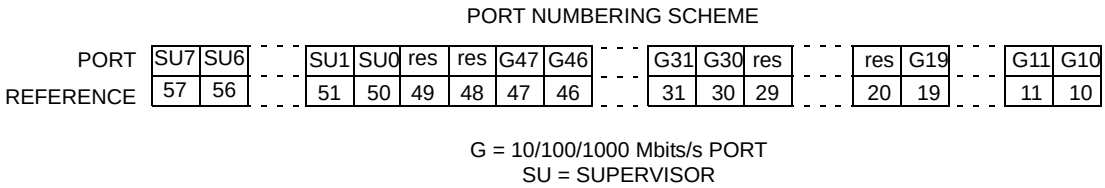


Figure 116. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Time_Stamp_Table

Description: The MAC address table's time stamps.

Table 150. Layer_2_Time_Stamp_Table Register Parameters

Parameter	Value
Base Address	0x000b_8000
Register Size	32768
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	8192
Record Spacing	4

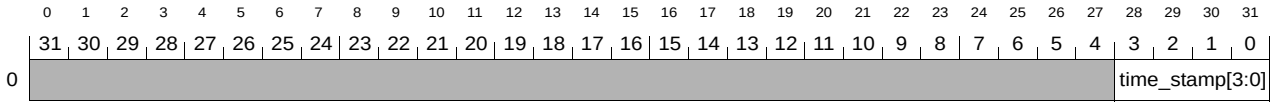


Figure 117. Layer_2_Time_Stamp_Table Register Diagram

Table 151. Layer_2_Time_Stamp_Table Field Parameters

Field Name	Parameter	Description
time_stamp[3:0]	Mode = R/W Offset = 0.28 Instances = 1	The time stamp value for the corresponding MAC source address.

This table holds the time stamps for the MAC address table. Whenever an address in the MAC address table is seen in a received packet as a source address and the received packet's source port matches the source port information in the address table, the time stamp value in this table that corresponds to that MAC address table entry is updated with the current time value. The current time value is established by `Layer_2_Current_Time`.

Appendix A: Registers (continued)

Layer_2_User_Port

Description: A user-specified TCP destination port for snooping purposes.

Table 152. Layer_2_User_Port Register Parameters

Parameter	Value
Base Address	0x000c_4678
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

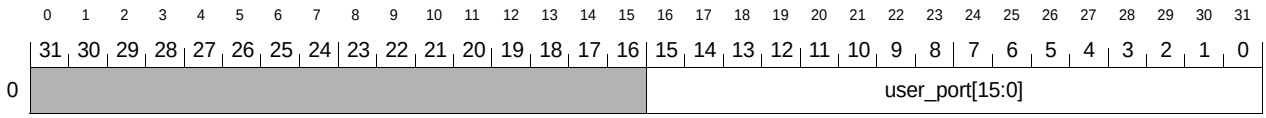


Figure 118. Layer_2_User_Port Register Diagram

Table 153. Layer_2_User_Port Field Parameters

Field Name	Parameter	Description
user_port[15:0]	Mode = R/W Offset = 0.16 Instances = 1	The user-specified TCP destination port to be snooped.

A single TCP destination port may be specified for snooping purposes. If so enabled, any packet with a TCP destination port value that matches the value here is copied to the port specified by `Layer_2_User_Port_Snooping_Port`.

Appendix A: Registers (continued)

Layer_2_User_Port_Snooping_Port

Description: Specifies the supervisor port (queue) to that user-port snooped packets are forwarded.

Table 154. Layer_2_User_Port_Snooping_Port Register Parameters

Parameter	Value
Base Address	0x000c_467c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

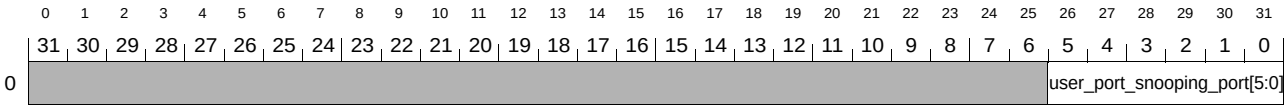


Figure 119. Layer_2_User_Port_Snooping_Port Register Diagram

Table 155. Layer_2_User_Port_Snooping_Port Field Parameters

Field Name	Parameter	Description
user_port_snooping_port[5:0]	Mode = R/W Offset = 0.26 Instances = 1	The port to which packets with a user-specified TCP destination port value are copied.

If the TCP destination port number of the received packet matches the value in `Layer_2_User_Port` and `Layer_2_Mode.user_port_snoop_en` is asserted, then the port specified by the value in `user_port_snooping_port[5:0]` is added to the packet's destination port map.

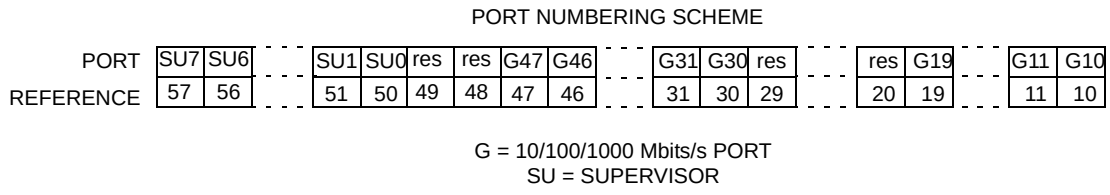


Figure 120. Port Numbering Scheme

Appendix A: Registers (continued)

Layer_2_Vlan_Mask_Table

Description: Enables the limiting of destinations based on source VLAN.

Table 156. Layer_2_Vlan_Mask_Table Register Parameters

Parameter	Value
Base Address	0x000c_3800
Register Size	2048
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	256
Record Spacing	8

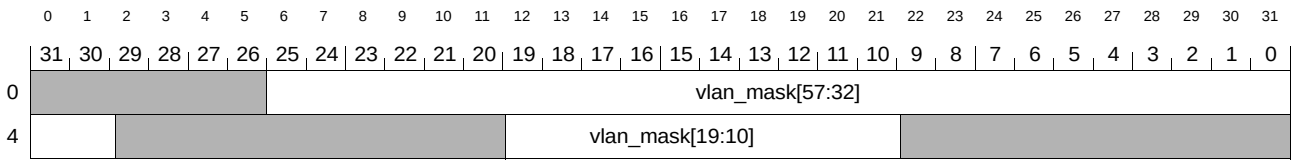


Figure 121. Layer_2_Vlan_Mask_Table Register Diagram

Table 157. Layer_2_Vlan_Mask_Table Field Parameters

Field Name	Parameter	Description
vlan_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The VLAN mask.

The received packet's 8-bit VLAN index is used to retrieve a port mask from this table. This mask is then used to eliminate destinations from the packet's destination port map.

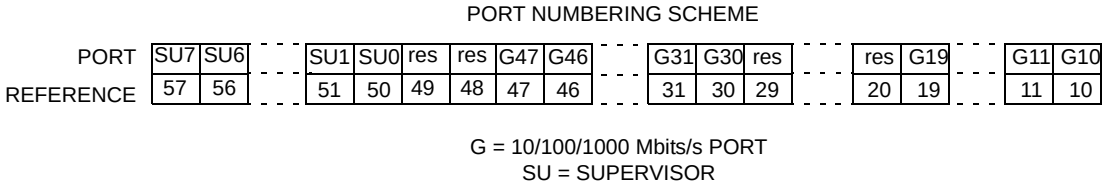


Figure 122. Port Numbering Scheme

Appendix A: Registers (continued)

Mac_Global_Mode

Description: Sets basic shared operating modes for the Ethernet MACs.

Table 158. Mac_Global_Mode Register Parameters

Parameter	Value
Base Address	0x000c_8310
Register Size	16
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	1
Record Spacing	NA

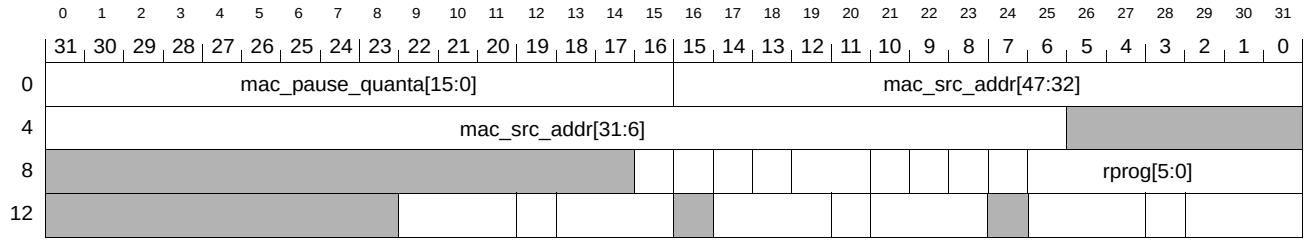


Figure 123. Mac_Global_Mode Register Diagram

Table 159. Mac_Global_Mode Field Parameters

Field Name	Parameter	Description
mac_pause_quanta[15:0]	Mode = R/W Offset = 0.0 Instances = 1	The pause quanta value to be inserted into pause control packets.
mac_src_addr[47:6]	Mode = R/W Offset = 0.16 Instances = 1	The upper 42 bits of the MAC source addresses. The least significant 6 bits are hardwired per port.
termination_compensation_disable	Mode = R/W Offset = 8.15 Instances = 1	Asserting this bit disables the SGMII termination compensation function for all channels. For normal operation, this bit is deasserted.
auto_negotiation_speed_up	Mode = R/W Offset = 8.16 Instances = 1	Asserting this bit causes the SGMII autonegotiation process to be sped up. For normal operation, this bit is deasserted. This bit has no effect on the autonegotiation process executed by the external PHY on the network.
prbs_en	Mode = R/W Offset = 8.17 Instances = 1	This bit is asserted to enable the PRBS test mode for all SGMII channels. The PRBS test runs for as long as this bit is asserted. For normal operation, this bit is deasserted.

Appendix A: Registers (continued)

Mac_Global_Mode (continued)

Table 159. Mac_Global_Mode Field Parameters (continued)

Field Name	Parameter	Description
inject_offset_polarity	Mode = R/W Offset = 8.18 Instances = 1	This bit establishes the polarity of the offset specified by `inject_offset{0..2}`. 0 = positive polarity offset. 1 = negative polarity offset.
loopback_delay_select[1:0]	Mode = R/W Offset = 8.19 Instances = 1	Selects a delay value when the SGMII is operating in a serial loopback mode. 00_2 = 400 ps. 01_2 = 100 ps. 10_2 = –100 ps.
loopback_amplitude_select	Mode = R/W Offset = 8.21 Instances = 1	This bit selects one of two amplitudes for serial loopback of the SGMII channels. 0 = 400 mV. 1 = 100 mV.
inject_offset_2	Mode = R/W Offset = 8.22 Instances = 1	Asserting this bit causes an offset current of level 2 to be injected into the SGMII data.
inject_offset_1	Mode = R/W Offset = 8.23 Instances = 1	Asserting this bit causes an offset current of level 1 to be injected into the SGMII data.
inject_offset_0	Mode = R/W Offset = 8.24 Instances = 1	Asserting this bit causes an offset current of level 0 to be injected into the SGMII data.
sw_reset_control	Mode = R/W Offset = 8.25 Instances = 1	Provides a software reset control to the device.
rprog[5:0]	Mode = R/W Offset = 8.26 Instances = 1	PLL parameter. Bits 4, 2, 1, and 0 of this value are inverted between the register and the PLL. This is done so that the reset value of 0 appears at the PLL as 01_0111_2.
set_resistor_value_{0..2}[2:0]	Mode = R/W Offset = 12.9 Instances = 3 Spacing = 0.8	This value is used in the SGMII termination block if the corresponding `force_resistor` bit is asserted. There are three independent termination control groups serving the 28 SGMII ports.
force_resistor_{0..2}	Mode = R/W Offset = 12.12 Instances = 3 Spacing = 0.8	When the SGMII termination compensation function is disabled, a high on this bit enables the application of the termination value specified in `set_resistor_value[2:0]`. There are three independent termination control groups serving the 28 SGMII ports.
resistor_compensation_status_{0..2}[2:0]	Mode = RO Offset = 12.13 Instances = 3 Spacing = 0.8	This field returns the resistor compensation results. There are three independent termination control groups serving the 28 SGMII ports.

Appendix A: Registers (continued)

Mac_Mode_{0..4}

Description: Sets basic operating modes for the Ethernet MACs.

Table 160. Mac_Mode_{0..4} Register Parameters

Parameter	Value
Base Address	0x000c_8100
Register Size	40
Register Instances	5
Register Spacing	128
Record Size	4
Record Instances	10
Record Spacing	4

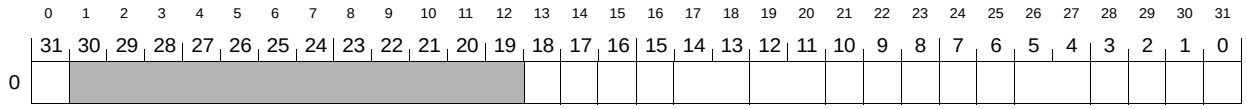


Figure 124. Mac_Mode_{0..4} Register Diagram

Table 161. Mac_Mode_{0..4} Field Parameters

Field Name	Parameter	Description
port_en_{}{0..9}	Mode = R/W Offset = 0.0 Instances = 1	Enables the corresponding Ethernet port. When a `port_en` bit is false, all activity of the associated Ethernet port is disabled.
good_link_{}{0..9}	Mode = RO Offset = 0.13 Instances = 1	This read-only bit is asserted whenever a good link status exists for the corresponding SGMII channel. Forcing a good link status by asserting `force_good_link` also causes `good_link` to be asserted.
full_duplex_{}{0..9}	Mode = RO Offset = 0.14 Instances = 1	This read-only bit is asserted whenever a full-duplex status exists for the corresponding SGMII channel. Forcing a full-duplex status by asserting `force_full_duplex` also causes `full_duplex` to be asserted.
offset_compensation_disable_{}{0..9}	Mode = R/W Offset = 0.15 Instances = 1	This bit is asserted to disable the SGMII offset compensation circuitry. For normal operation, this bit is deasserted.
power_down_{}{0..9}	Mode = R/W Offset = 0.16 Instances = 1	Asserting this bit powers down the associated SGMII interface module. For normal operation, this bit is deasserted.

Appendix A: Registers (continued)

Mac_Mode_{0..4} (continued)

Table 161. Mac_Mode_{0..4} Field Parameters (continued)

Field Name	Parameter	Description
input_select_{}{0..9}[1:0]	Mode = R/W Offset = 0.17 Instances = 1	This field is used to select the source of receive data and receive clocks for those Ethernet ports that support only a SerDes interface. This field is encoded as follows: 00_2 = data disabled, SGMII clock 01_2 = reserved 10_2 = SerDes data and clock 11_2 = data disabled, SGMII clock
speed_mode_force_{}{0..9}[1:0]	Mode = R/W Offset = 0.19 Instances = 1	Causes the speed mode of the associated 1 Gbit/s link to be forced to the specified setting. This field is only valid when `good_link_force` is asserted. `speed_mode_force`[1:0] is defined as follows: 00_2 = 10 Mbits/s 01_2 = 100 Mbits/s 10_2 = 1 Gbit/s 11_2 = reserved
good_link_force_{}{0..9}	Mode = R/W Offset = 0.21 Instances = 1	When asserted, associated link is forced to operate as if it has detected a valid link signal. The assertion of this bit also enables `speed_mode_force[1:0]` and `full_duplex_force`.
full_duplex_force_{}{0..9}	Mode = R/W Offset = 0.22 Instances = 1	When asserted, the associated PHY is forced into a full-duplex mode of operation. This signal is only valid when `good_link_force` is asserted.
auto_negotiate_en_{}{0..9}	Mode = R/W Offset = 0.23 Instances = 1	When asserted, autonegotiation is enabled for the associate PHY.
restart_auto_negotiation_{}{0..9}	Mode = R/W Offset = 0.24 Instances = 1	Autonegotiation may be forced to be restarted by asserting and then immediately deasserting this bit. The rising edge (0 to 1 transition) is detected in order to force a single autonegotiation.
gmac_tx_flush_{}{0..9}	Mode = R/W Offset = 0.25 Instances = 1	When asserted, the associated PHY accepts transmit data from `packet_buffer` at the maximum data rate and then discards it without transmitting it. Changes to this mode bit only take effect between transmit packets.
gmac_port_speed_{}{0..9}[1:0]	Mode = RO Offset = 0.26 Instances = 1	Indicates the port speed of the associated PHY as follows: 00_2 = 10 Mbits/s 01_2 = 100 Mbits/s 10_2 = 1 Gbit/s 11_2 = reserved

Appendix A: Registers (continued)

Mac_Mode_{0..4} (continued)

Table 161. Mac_Mode_{0..4} Field Parameters (continued)

Field Name	Parameter	Description
gmac_port_loopback_en_{}{0..9}	Mode = R/W Offset = 0.28 Instances = 1	When asserted, all transmit packets are looped back to the receive path by the associated SGMII interface. Network reception is disabled during loopback. Changes to this mode bit take effect immediately. This may cause the transmission or reception of one or more defective packets.
gmac_rx_en_{}{0..9}	Mode = R/W Offset = 0.29 Instances = 1	Enables the reception of Ethernet packets. When false, only MAC control packets are received.
gmac_flow_control_initiate_en_{}{0..9}	Mode = R/W Offset = 0.30 Instances = 1	This bit enables the initiation of flow control actions on the part of the Ethernet MAC in response to flow control indications from `packet_buffer`. If this bit is deasserted, then the corresponding Ethernet MAC never takes any flow control actions. In revision C, when `input_select_{}{0..9}[1:0]` = 0x2 and `auto_negotiate_en_{}{0..9}` = 1, this field becomes the ASM_DIR (PS2) bit for 1000BASE-X autonegotiation.
gmac_rx_pause_en_{}{0..9}	Mode = R/W Offset = 0.31 Instances = 1	This bit must be asserted for the corresponding Ethernet MAC to react to the reception of MAC flow control packets. In revision C, when `input_select_{}{0..9}[1:0]` = 0x2 and `auto_negotiate_en_{}{0..9}` = 1, this field becomes the PAUSE (PS1) bit for 1000BASE-X autonegotiation.

Note: Mac_Mode_{0} and Mac_Mode_{2} are reserved and are not used.

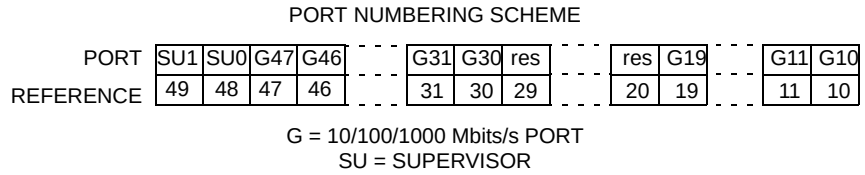


Figure 125. Port Numbering Scheme

Appendix A: Registers (continued)

Mac_Status_{0..4}

Description: Indicates fundamental status for the Ethernet MACs.

Table 162. Mac_Status_{0..4} Register Parameters

Parameter	Value
Base Address	0x000c_8150
Register Size	4
Register Instances	5
Register Spacing	128
Record Size	4
Record Instances	1
Record Spacing	NA

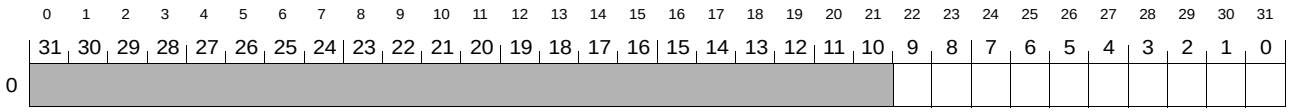


Figure 126. Mac_Status_{0..4} Register Diagram

Table 163. Mac_Status_{0..4} Field Parameters

Field Name	Parameter	Description
prbs_error_{0..9}{}	Mode = R/W Offset = 0.22 Instances = 10 Spacing = 0.1	This bit is asserted to indicate that an error has occurred during PRBS testing of the corresponding SGMII channel. This bit is cleared by writing a one to its location. Writing a zero has no effect.

Note: Mac_Status_{0} and Mac_Status_{2} are reserved and are not used.

PORT NUMBERING SCHEME

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 127. Port Numbering Scheme

Appendix A: Registers (continued)

Mdio_Control

Description: Provides supervisor control of the MDIO interfaces.

Table 164. Mdio_Control Register Parameters

Parameter	Value
Base Address	0x000c_8300
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

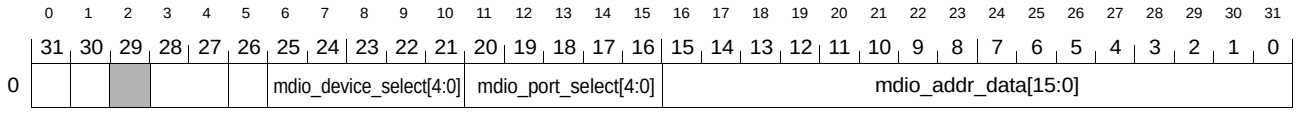


Figure 128. Mdio_Control Register Diagram

Table 165. Mdio_Control Field Parameters

Field Name	Parameter	Description
mdio_busy	Mode = RO Offset = 0.0 Instances = 1	This bit is asserted during the period that a previously initiated MDIO access cycle is running. `mdio_busy` is asserted by a write of the access cycle's addressing parameters. If a read cycle was posted, then the read data is valid immediately after the deassertion of mdio_busy. Reads of `mdio_data[15:0]` and writes to any field of `Mdio_Control` are not allowed while `mdio_busy` is asserted.
suppress_preamble	Mode = R/W Offset = 0.1 Instances = 1	When asserted, the MDIO controller suppresses the preamble that normally precedes an MDIO command packet.

Appendix A: Registers (continued)

Mdio_Control (continued)

Table 165. Mdio_Control Field Parameters (continued)

Field Name	Parameter	Description
mdio_opcode[1:0]	Mode = R/W Offset = 0.3 Instances = 1	This field defines the type of operation being carried out by a write to this register. The opcodes are defined as follows: 00_2 = address 01_2 = write 10_2 = read incremental 11_2 = read Address: This opcode is utilized to establish the cycle's parameters. Write: The supervisor uses the 01_2 opcode to indicated that a write cycle is to be initiated. The write data is written to *m*`dio_addr_data[15:0]`. Read Incremental: This read operation initiates a read from the next (+1) address; the current address having been established by a preceding address operation. Aside from the autoincrement of the read address, this operation is identical to a normal read operation (opcode = 11_2). Read: Reads are initiated by writing the read address information and the 11_2 opcode to this register. The resulting read data is made available via `mdio_addr_data[15:0]` after `mdio_busy` is deasserted by the MDIO interface controller.
mdio_interface_select	Mode = R/W Offset = 0.5 Instances = 1	Selects interface zero when deasserted and interface one when asserted.
mdio_device_select[4:0]	Mode = R/W Offset = 0.6 Instances = 1	Selects one of 32 devices on the selected MDIO interface.
mdio_port_select[4:0]	Mode = R/W Offset = 0.11 Instances = 1	Selects one of 32 ports on the selected MDIO device.
mdio_addr_data[15:0]	Mode = R/W Offset = 0.16 Instances = 1	The register address or data portal.

MDIO registers on the attached Ethernet PHY devices are accessed in two stages. The first stage sets up the cycle's parameters (addresses, device selects, modes, etc.), and the second stage accomplishes the data transfer.

For writes, the write data is simply written to offset 0 of this register with an opcode of 01_2. The initiation of a write causes `mdio_busy` to be asserted by the MDIO interface controller. This signal is deasserted automatically upon completion of the MDIO access cycle. No further MDIO access cycles may be initiated while `mdio_busy` is asserted.

For reads, the cycle's address parameters are written to initiate the access cycle. This action causes the assertion of `mdio_busy`. Once the read cycle is complete, `mdio_busy` is deasserted and the read data is posted to `mdio_addr_data[15:0]` where it may be accessed by the supervisor.

Appendix A: Registers (continued)

Mdio_Mode

Description: Defines the basic mode for the MDIO interfaces.

Table 166. Mdio_Mode Register Parameters

Parameter	Value
Base Address	0x000c_8308
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

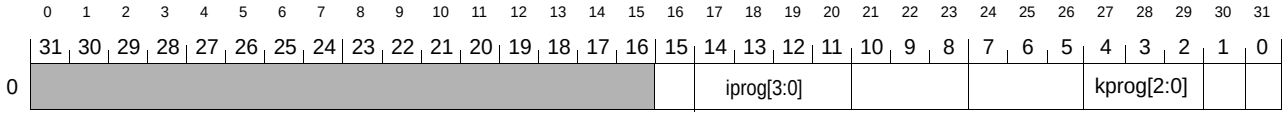


Figure 129. Mdio_Mode Register Diagram

Table 167. Mdio_Mode Field Parameters

Field Name	Parameter	Description
enbwp	Mode = R/W Offset = 0.16 Instances = 1	PLL parameter.
iprog[3:0]	Mode = R/W Offset = 0.17 Instances = 1	PLL parameter. iprog[2] is inverted between this register and the PLL so that the default reset value is 0100_2.
mdio_clk_offset[2:0]	Mode = R/W Offset = 0.21 Instances = 1	This field establishes the delay between the active edge of the MDIO clock and data.
mdio_clk_period[7:5]	Mode = R/W Offset = 0.24 Instances = 1	This field defines the half-clock period of the MDIO clock. mdio_clk_period[4:0] is hardwired to be equal to 31.
kprog[2:0]	Mode = R/W Offset = 0.27 Instances = 1	PLL parameter. kprog[0] is inverted between this register and the PLL.This has the effect of making the after-reset value equal to 0100_2.
accept_jumbo_frames	Mode = R/W Offset = 0.30 Instances = 1	When this bit is set, all ports may receive frames greater than 1522 bytes.
short_ifg_mode	Mode = R/W Offset = 0.31 Instances = 1	When this bit is set, 10 Mbits/s and 100 Mbits/s ports will transmit frames with an interframe gap of 11 bytes instead of 12 bytes. This bit should be cleared for normal operation.

Appendix A: Registers (continued)

Mdio_Status

Description: Provides supervisor control of the MDIO interfaces.

Table 168. Mdio_Status Register Parameters

Parameter	Value
Base Address	0x000c_8304
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

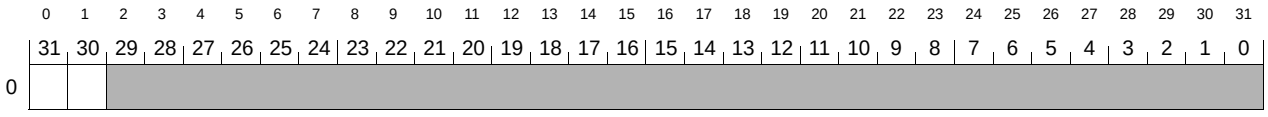


Figure 130. Mdio_Status Register Diagram

Table 169. Mdio_Status Field Parameters

Field Name	Parameter	Description
mdio_busy	Mode = RO Offset = 0.0 Instances = 1	This bit is asserted during the period that a previously initiated MDIO access cycle is running. `mdio_busy` is asserted by a write of the access cycle's addressing parameters. If a read cycle was posted, then the read data is valid immediately after the deassertion of mdio_busy. Reads of `mdio_data[15:0]` and writes to any field of `Mdio_Control` are not allowed while `mdio_busy` is asserted.
mdio_done	Mode = R/W Offset = 0.1 Instances = 1	This bit is asserted at the completion of an MDIO access cycle. The supervisor must write a zero to this bit's position in order to clear the indication.

This register provides basic status regarding MDIO access cycles.

`mdio_busy` provides the same information as the signal of the same name in `Mdio_Control`.

`mdio_done` is automatically asserted upon completion of an MDIO access cycle. This signal is also used as an interrupt-generating indication to the supervisor. The supervisor must write a zero to this bit's position to deassert the bit and cancel the indication.

Appendix A: Registers (continued)

Multicast_Rate_Accumulator

Description: Counts multicast and broadcast packets.

Table 170. Multicast_Rate_Accumulator Register Parameters

Parameter	Value
Base Address	0x000c_4814
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

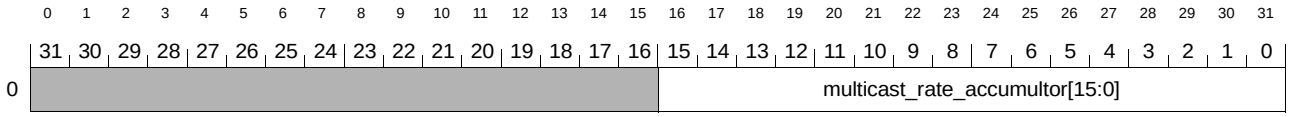


Figure 131. Multicast_Rate_Accumulator Register Diagram

Table 171. Multicast_Rate_Accumulator Field Parameters

Field Name	Parameter	Description
multicast_rate_accumulator[15:0]	Mode = R/W Offset = 0.16 Instances = 1	The multicast accumulator. Each multicast or broadcast packet received causes this counter in increment by one. This counter is decrement periodically by one with a period established by `Multicast_Rate_Decrement_Period`.

This leaky bucket counter is incremented by the reception of multicast and broadcast packets and is decremented by one on a fixed interval. That interval is defined by the contents of the `Multicast_Rate_Decrement_Period` register. Whenever the value of `multicast_rate_accumulator[15:0]` exceeds `Multicast_Rate_Limit`, the mask in `Multicast_Rate_Discard_Mask` is applied to the receive packet's destination map.

`Multicast_Rate_Accumulator` is never decremented through zero, nor allowed to exceed `Multicast_Rate_Limit` by more than one.

Appendix A: Registers (continued)

Multicast_Rate_Decrement_Period

Description: Defines the period between decrements of `Multicast_Rate_Accumulator`.

Table 172. Multicast_Rate_Decrement_Period Register Parameters

Parameter	Value
Base Address	0x000c_480c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

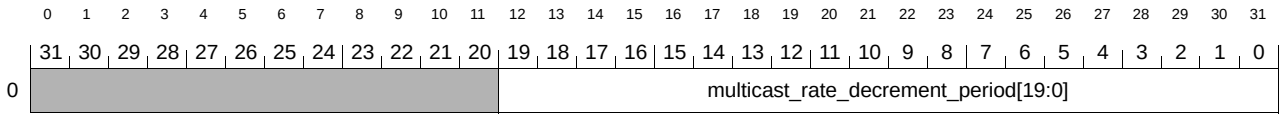


Figure 132. Multicast_Rate_Decrement_Period Register Diagram

Table 173. Multicast_Rate_Decrement_Period Field Parameters

Field Name	Parameter	Description
multicast_rate_decrement_period[19:0]	Mode = R/W Offset = 0.12 Instances = 1	Establishes the rate that `Multicast_Rate_Accumulator` is decremented.

This register specifies the number of 8 ns periods between decrements of `Multicast_Rate_Accumulator`. The minimum value of one results in a decrement once every 8 ns, or a total multicast/broadcast packet rate of 125,000,000 per second. This register's maximum value of 1,048,575 results in a total multicast/broadcast packet rate of 119 per second.

Appendix A: Registers (continued)

Multicast_Rate_Discard_Mask

Description: Disables those ports that should not receive packets that exceed the permitted multicast rate.

Table 174. Multicast_Rate_Discard_Mask Register Parameters

Parameter	Value
Base Address	0x000c_4800
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

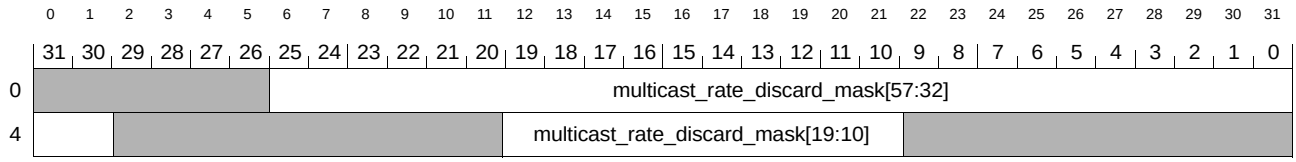


Figure 133. Multicast_Rate_Discard_Mask Register Diagram

Table 175. Multicast_Rate_Discard_Mask Field Parameters

Field Name	Parameter	Description
multicast_rate_discard_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1	The multicast rate discard mask value. Each bit in the mask corresponds to an Ethernet port or one of the supervisor's queues.

This mask is applied to the destination port map if the value of `Multicast_Rate_Accumulator` is greater than `Multicast_Rate_Limit`. This mask is only applied to multicast and broadcast packets.

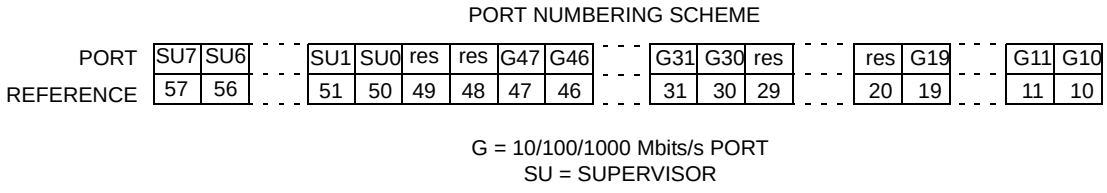


Figure 134. Port Numbering Scheme

Appendix A: Registers (continued)

Multicast_Rate_Limit

Description: Establishes the maximum value allowed in `Multicast_Rate_Accumulator`.

Table 176. Multicast_Rate_Limit Register Parameters

Parameter	Value
Base Address	0x000c_4808
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

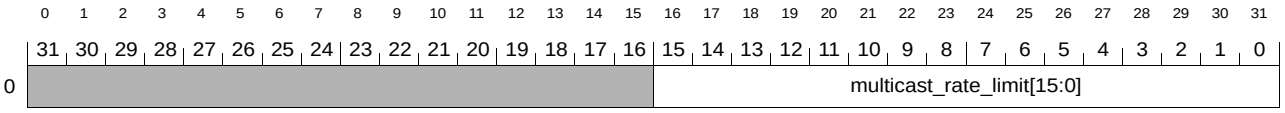


Figure 135. Multicast_Rate_Limit Register Diagram

Table 177. Multicast_Rate_Limit Field Parameters

Field Name	Parameter	Description
multicast_rate_limit[15:0]	Mode = R/W Offset = 0.16 Instances = 1	The upper limit permitted by `Multicast_Rate_Accumulator`.

This register establishes the maximum value permitted in `Multicast_Rate_Accumulator`. If `Multicast_Rate_Accumulator` is greater than `Multicat_Rate_Limit`, then the discarding of multicast packets is enabled. The maximum value allowed in `Multcast_Rate_Limit` is FFFE16.

The actual multicast rate is established via the `Multicast_Rate_Decrement_Period` register. `Multicast_Rate_Limit` merely establishes this function's tolerance of bursts of multicast and broadcast packets. A larger value of `multicast_rate_limit[15:0]` allows longer bursts of multicast and broadcast packet before the discarding of these packets kicks in.

Appendix A: Registers (continued)

Multicast_Rate_Limit_Events

Description: An accounting of the number of occurrences of multicast discards.

Table 178. Multicast_Rate_Limit_Events Register Parameters

Parameter	Value
Base Address	0x000c_4810
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

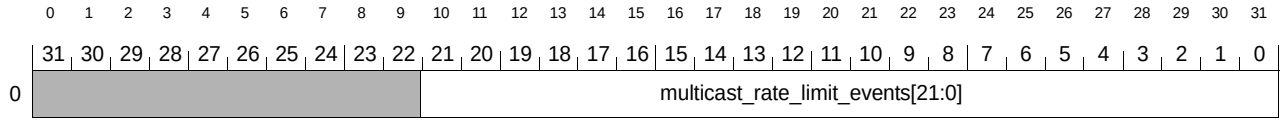


Figure 136. Multicast_Rate_Limit_Events Register Diagram

Table 179. Multicast_Rate_Limit_Events Field Parameters

Field Name	Parameter	Description
multicast_rate_limit_events[21:0]	Mode = R/W Offset = 0.10 Instances = 1	A statistics counter that track multicast discards.

This counter is incremented by one each time a multicast packet is received while `Multicast_Rate_Accumulator` is greater than `Multicast_Rate_Limit` and `Multicast_Rate_Limit` is greater than zero.

Note: The application of the multicast rate discard mask may not result in the reduction of the number of destinations for a multicast packet. For example, BPDU packets ordinarily should not be discarded by this function and `Multicast_Rate_Discard_Mask` is normally configured to avoid masking the supervisor's BPDU queue. However, the reception of a BPDU packet (a multicast packet) may cause the application of the mask (to no effect) and the incrementing of this counter.

Appendix A: Registers (continued)

Multicast_Rate_Mode

Description: Mode bits for the multicast rate limiting function.

Table 180. Multicast_Rate_Mode Register Parameters

Parameter	Value
Base Address	0x000c_4818
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

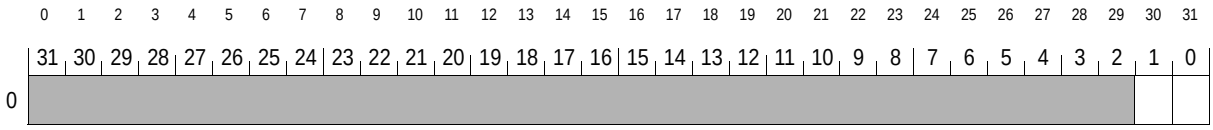


Figure 137. Multicast_Rate_Mode Register Diagram

Table 181. Multicast_Rate_Mode Field Parameters

Field Name	Parameter	Description
broadcast_rate_limit_en	Mode = R/W Offset = 0.30 Instances = 1	When asserted, the multicast rate-limiting function is enabled and applied to broadcast packets. When this bit is deasserted, `Multicast_Rate_Limit_Discard_Mask` is never applied and `Multicast_Rate_Limit_Events` is never incremented for broadcast packets.
multicast_rate_limit_en	Mode = R/W Offset = 0.31 Instances = 1	When asserted, the multicast rate-limiting function is enabled and applied to multicast packets. When this bit is deasserted, `Multicast_Rate_Limit_Discard_Mask` is never applied and `Multicast_Rate_Limit_Events` is never incremented for multicast packets.

Appendix A: Registers (continued)

Packet_Buffer_Acl_Deny_Mask

Description: Identifies those ports to be eliminated as destinations based on an ACL denial.

Table 182. Packet_Buffer_Acl_Deny_Mask Register Parameters

Parameter	Value
Base Address	0x000c_bc40
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

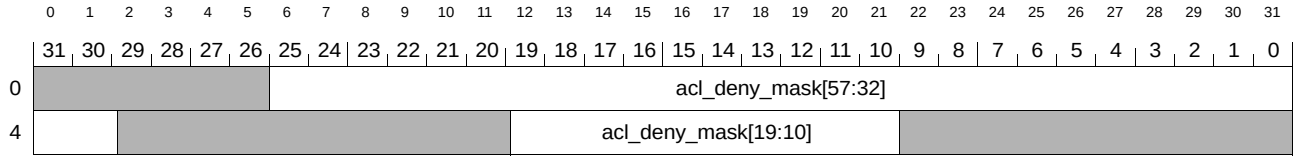


Figure 138. Packet_Buffer_Acl_Deny_Mask Register Diagram

Table 183. Packet_Buffer_Acl_Deny_Mask Field Parameters

Field Name	Parameters	Description
acl_deny_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The ACL denial mask.

If the ACL look-up returns a deny indication, then the ports identified by this mask are eliminated as destinations for the packet. Asserted bits in `acl_deny_mask[57:10]` identify those ports that are eliminated (masked) by this function. Deasserted bits in `acl_deny_mask[57:10]` have no effect on the packet's destination port map.

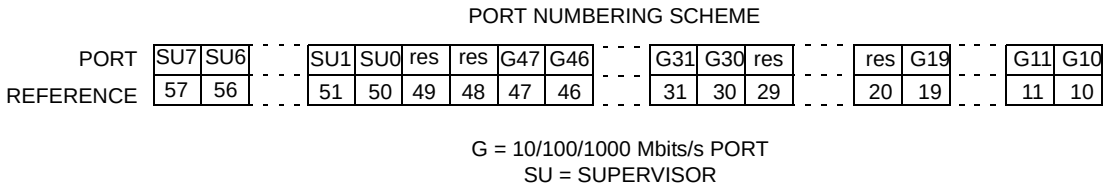


Figure 139. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Acl_Log_Port

Description: Identifies the ACL logging port.

Table 184. Packet_Buffer_Acl_Log_Port Register Parameters

Parameter	Value
Base Address	0x000c_bc58
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

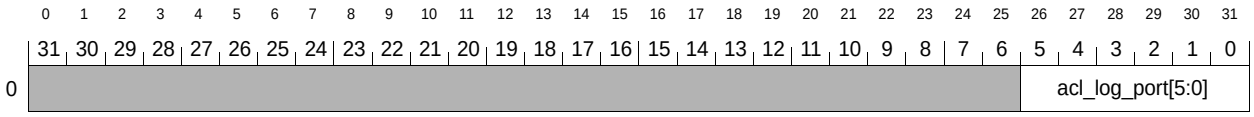


Figure 140. Packet_Buffer_Acl_Log_Port Register Diagram

Table 185. Packet_Buffer_Acl_Log_Port Field Parameters

Field Name	Parameters	Description
acl_log_port[5:0]	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	The ACL logging port number.

If the ACL look-up returns a log indication, then the port identified by this value is added to the packet's destination port map. Most typically, one of the supervisor's queues (ports 50 through 57) are designated as the ACL logging port.

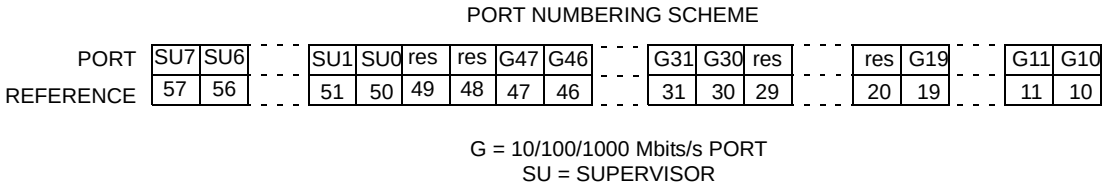


Figure 141. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Allocated_Buffer_Count

Description: Provides a real-time indication of the number of allocated buffers.

Table 186. Packet_Buffer_Channel_Congestion_Threshold Register Parameters

Parameter	Value
Base Address	0x000c_b900
Register Size	208
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	50
Record Spacing	4

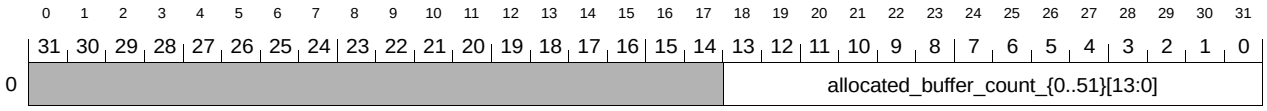


Figure 142. Packet_Buffer_Allocated_Buffer_Count Register Diagram

Table 187. Packet_Buffer_Allocated_Buffer_Count Field Parameters

Field Name	Parameter	Description
allocated_buffer_count[13:0]	Mode = R/W Offset = 0.18 Instances = 1	The number of buffers allocated to the corresponding port.

This register provides a real-time indication of the number of buffers that are allocated per port.

PORT NUMBERING SCHEME (TABLE INDEXING)

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 143. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Channel_Congestion_Threshold

Description: Per-channel congestion thresholds.

Table 188. Packet_Buffer_Channel_Congestion_Threshold Register Parameters

Parameter	Value
Base Address	0x000c_b800
Register Size	200
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	50
Record Spacing	4

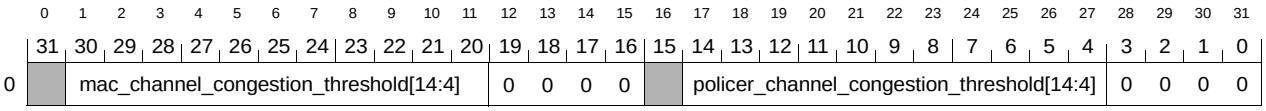


Figure 144. Packet_Buffer_Channel_Congestion_Threshold Register Diagram

Table 189. Packet_Buffer_Channel_Congestion_Threshold Field Parameters

Field Name	Parameter	Description
mac_channel_congestion_threshold[14:4]	Mode = R/W Offset = 0.1 Instances = 1	When the number of buffers allocated to the corresponding channel equals or exceeds this value, the channel is considered congested. The number of buffers allocated must drop below `congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]` for the channel's state to change from congested to not congested. The results of this congestion test are provided to the Ethernet MACs.
policer_channel_congestion_threshold[14:4]	Mode = R/W Offset = 0.17 Instances = 1	When the number of buffers allocated to the corresponding channel equals or exceeds this value, the channel is considered congested. The number of buffers allocated must drop below `congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]` for the channel's state to change from congested to not congested. The results of this congestion test are provided to the policers within the `packet_processor` modules.

Appendix A: Registers (continued)

Packet_Buffer_Channel_Congestion_Threshold (continued)

PORT NUMBERING SCHEME (TABLE INDEXING)

PORT	SU1	SU0	G47	G46	G45	G44		G23	G22	G21	G20
REFERENCE	49	48	47	46	45	44		23	22	21	20

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 145. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Crc_Error_Mask

Description: Identifies those ports to be eliminated as destinations in the event of a receive CRC error.

Table 190. Packet_Buffer_Crc_Error_Mask Register Parameters

Parameter	Value
Base Address	0x000c_bc48
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

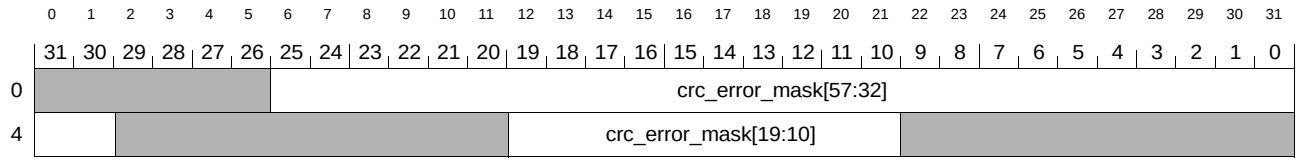


Figure 146. Packet_Buffer_Crc_Error_Mask Register Diagram

Table 191. Packet_Buffer_Crc_Error_Mask Field Parameters

Field Name	Parameter	Description
crc_error_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The CRC error mask.

If the packet is received with a CRC error (or some other form of receive error that indicates corrupted or unreliable packet data), then the ports identified by this mask are eliminated as destinations for the packet. Asserted bits in `crc_error_mask[57:10]` identify those ports that are eliminated (masked) by this function. Deasserted bits in `crc_error_mask[57:10]` have no effect on the packet's destination port map.

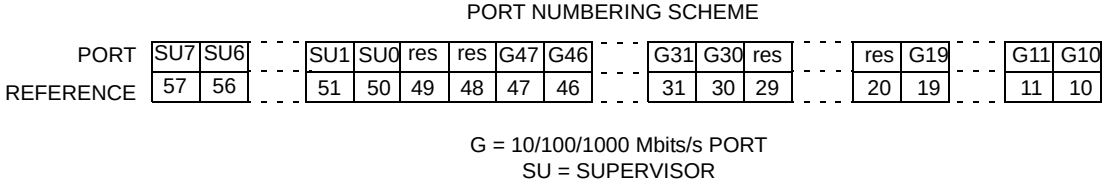


Figure 147. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Descriptor_Congestion_Threshold

Description: Global descriptor congestion thresholds.

Table 192. Packet_Buffer_Descriptor_Congestion_Threshold Register Parameters

Parameter	Value
Base Address	0x000c_bc5c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

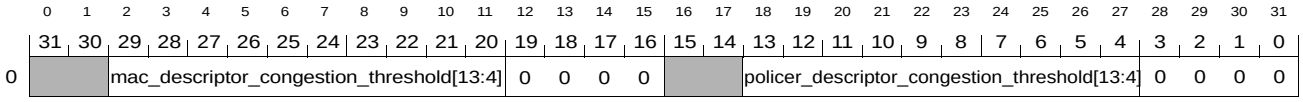


Figure 148. Packet_Buffer_Descriptor_Congestion_Threshold Register Diagram

Table 193. Packet_Buffer_Descriptor_Congestion_Threshold Field Parameters

Field Name	Parameter	Description
mac_descriptor_congestion_threshold[13:4]	Mode = R/W Offset = 0.2 Instances = 1	When the number of descriptors allocated equals or exceeds this value, the packet buffer is considered congested. The number of descriptors allocated must drop below `congestion_threshold[13:4]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. The results of this congestion test are provided to the Ethernet MACs. **Note:** For proper operation, the maximum value for this field should be 0x160 or less.
policer_descriptor_congestion_threshold[13:4]	Mode = R/W Offset = 0.18 Instances = 1	When the number of descriptors allocated equals or exceeds this value, the packet buffer is considered congested. The number of descriptors allocated must drop below `congestion_threshold[13:4]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. The results of this congestion test are provided to the policers within the `packet_processor` modules. **Note:** For proper operation, the maximum value for this field should be 0x160 or less.

Packet descriptors are managed in four groups. The global thresholds defined by this register apply to all four groups. If any one of the four groups exceed an allocation threshold, then the threshold is considered crossed by all four groups.

Appendix A: Registers (continued)

Packet_Buffer_Descriptor_Usage_Count

Description: Provides a real-time indication of the number of descriptors allocated to each port group.

Table 194. Packet_Buffer_Descriptor_Usage_Count Register Parameters

Parameter	Value
Base Address	0x000c_bb60
Register Size	16
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	4
Record Spacing	NA

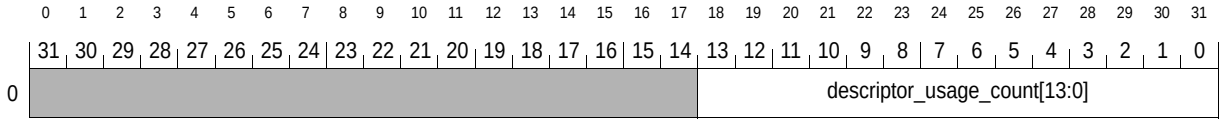


Figure 149. Packet_Buffer_Descriptor_Usage_Count Register Diagram

Table 195. Packet_Buffer_Descriptor_Usage_Count Field Parameters

Field Name	Parameter	Description
descriptor_usage_count[13:0]	Mode = R/W Offset = 0.18 Instances = 1	The number of descriptors allocated to the corresponding port group.

This register provides a real-time indication of the number of descriptors that have been allocated to one of four port groups. Ports are grouped together as follows:

Table 196. Port/Group Associations

Group	Ports
0	20, 24, 28, 32, 36, 40, 44
1	21, 25, 29, 33, 37, 41, 45
2	22, 26, 30, 34, 38, 42, 46
3	23, 27, 31, 35, 39, 43, 47

Appendix A: Registers (continued)

Packet_Buffer_Discard_Mask

Description: Identifies those ports to be eliminated as destinations in the event of a discard indication.

Table 197. Packet_Buffer_Discard_Mask Register Parameters

Parameter	Value
Base Address	0x000c_bc50
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	1
Record Spacing	NA

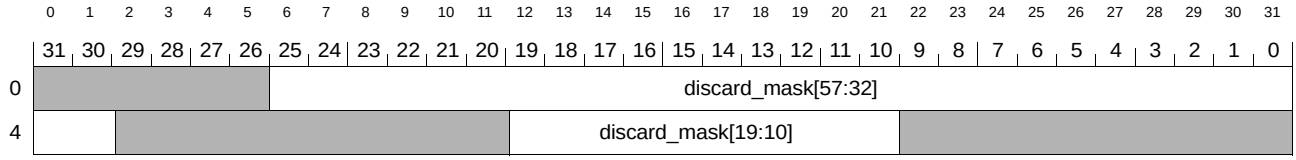


Figure 150. Packet_Buffer_Discard_Mask Register Diagram

Table 198. Packet_Buffer_Discard_Mask Field Parameters

Field Name	Parameter	Description
discard_mask[57:10]	Mode = R/W Offset = 0.6 Instances = 1 Reset = 0	The discard mask.

If the packet is marked for discard by some ingress process, then the ports identified by this mask are eliminated as destinations for the packet. Asserted bits in `discard_mask[57:10]` identify those ports that are eliminated (masked) by this function. Deasserted bits in `discard_mask[57:10]` have no effect on the packet's destination port map.

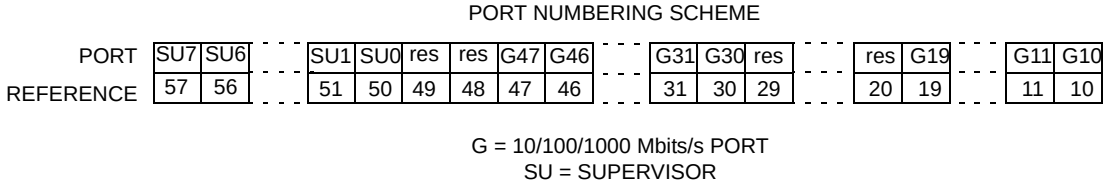


Figure 151. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Free_Buffer_Control

Description: Free buffer list initialization controls.

Table 199. Packet_Buffer_Free_Buffer_Control Register Parameters

Parameter	Value
Base Address	0x000c_bb70
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

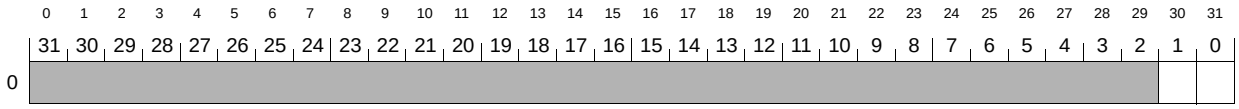


Figure 152. Packet_Buffer_Free_Buffer_Control Register Diagram

Table 200. Packet_Buffer_Free_Buffer_Control Field Parameters

Field Name	Parameter	Description
free_buffer_initialization_start	Mode = WO Offset = 0.30 Instances = 1	Writing a one to this bit starts the initialization process for the free buffer list.
free_buffer_initialization_done	Mode = RO Offset = 0.31 Instances = 1	This bit indicates that the initialization of the free buffer list is complete. This bit is deasserted during the initialization process.

Free buffer list initialization controls.

Appendix A: Registers (continued)

Packet_Buffer_Free_Buffer_Count_Even

Description: Provides a real-time indication of the number of free buffers.

Table 201. Packet_Buffer_Free_Buffer_Count_Even Register Parameters

Parameter	Value
Base Address	0x000c_bb84
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

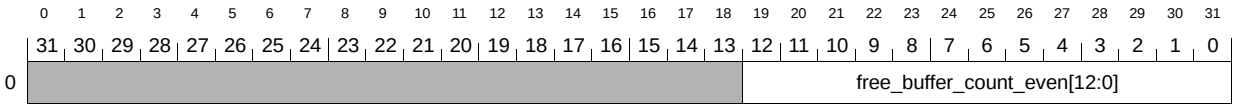


Figure 153. Packet_Buffer_Free_Buffer_Count_Even Register Diagram

Table 202. Packet_Buffer_Free_Buffer_Count_Even Field Parameters

Field Name	Parameter	Description
free_buffer_count_even[12:0]	Mode = WO Offset = 0.19 Instances = 1	The number of buffers available for allocation to storage.

This register provides a real-time indication of the number of buffers that reside on the free list. Smaller numbers read here indicate greater levels of congestion.

This register corresponds to all even-numbered buffers.

Appendix A: Registers (continued)

Packet_Buffer_Free_Buffer_Count_Odd

Description: Provides a real-time indication of the number of free buffers.

Table 203. Packet_Buffer_Free_Buffer_Count_Odd Register Parameters

Parameter	Value
Base Address	0x000c_bb88
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

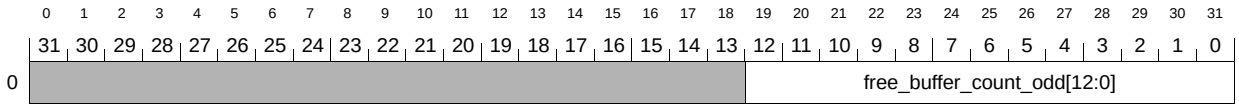


Figure 154. Packet_Buffer_Free_Buffer_Count_Even Register Diagram

Table 204. Packet_Buffer_Free_Buffer_Count_Even Field Parameters

Field Name	Parameter	Description
free_buffer_count_odd[12:0]	Mode = WO Offset = 0.19 Instances = 1	The number of buffers available for allocation to storage.

This register provides a real-time indication of the number of buffers that reside on the free list. Smaller numbers read here indicate greater levels of congestion.

This register corresponds to all odd-numbered buffers.

Appendix A: Registers (continued)

Packet_Buffer_Free_Descriptor_Control

Description: Free buffer list initialization controls.

Table 205. Packet_Buffer_Free_Descriptor_Control Register Parameters

Parameter	Value
Base Address	0x000c_bb74
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

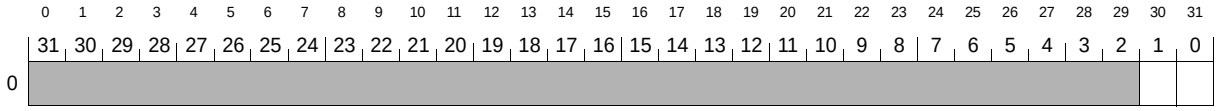


Figure 155. Packet_Buffer_Free_Descriptor_Control Register Diagram

Table 206. Packet_Buffer_Free_Descriptor_Control Field Parameters

Field Name	Parameter	Description
free_descriptor_initialization_start	Mode = WO Offset = 0.30 Instances = 1	Writing a one to this bit starts the initialization process for the free descriptor list.
free_descriptor_initialization_done	Mode = RO Offset = 0.31 Instances = 1	This bit indicates that the initialization of the free descriptor list is complete. This bit is deasserted during the initialization process.

Free descriptor list initialization controls.

Appendix A: Registers (continued)

Packet_Buffer_Global_Congestion_Threshold

Description: Global congestion thresholds.

Table 207. Packet_Buffer_Global_Congestion_Threshold Register Parameters

Parameter	Value
Base Address	0x000c_bc60
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

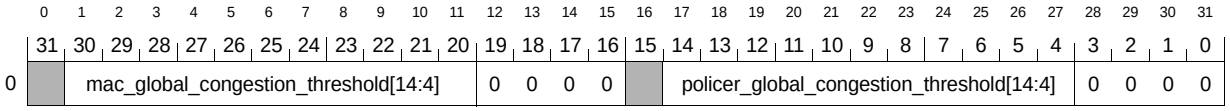


Figure 156. Packet_Buffer_Global_Congestion_Threshold Register Diagram

Table 208. Packet_Buffer_Global_Congestion_Threshold Field Parameters

Field Name	Parameter	Description
mac_global_congestion_threshold[14:4]	Mode = R/W Offset = 0.1 Instances = 1	When the total number of buffers (128 bytes) allocated equals or exceeds this value, the packet buffer is considered congested. The number of buffers allocated must drop below `congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. The results of this congestion test are provided to the Ethernet MACs. **Note:** For proper operation, the maximum value for this field should be 0x1A8 or less.
policer_global_congestion_threshold[14:4]	Mode = R/W Offset = 0.17 Instances = 1	When the total number of buffers (128 bytes) allocated equals or exceeds this value, the packet buffer is considered congested. The number of buffers allocated must drop below `congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. The results of this congestion test are provided to the policers within the `packet_processor` modules. **Note:** For proper operation, the maximum value for this field should be 0x1A8 or less.

Packet buffer congestion thresholds.

Appendix A: Registers (continued)

Packet_Buffer_Ind

Description: Provides supervisor indications from **packet_buffer**.

Table 209. Packet_Buffer_Ind Register Parameters

Parameter	Value
Base Address	0x000c_bb40
Register Size	32
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

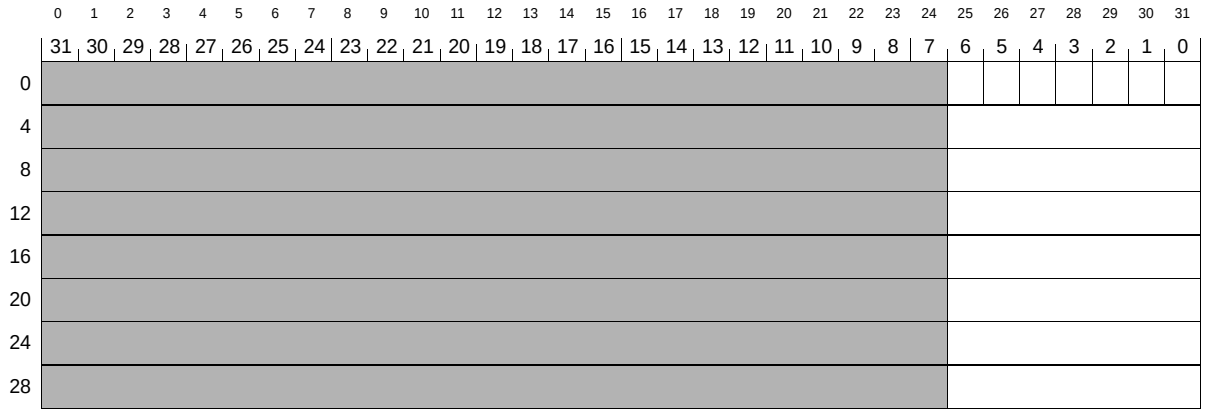


Figure 157. Packet_Buffer_Ind Register Diagram

Table 210. Packet_Buffer_Ind Field Parameters

Field Name	Parameter	Description
packet_buffer_parity_error	Mode = R/W Offset = 0.25 Instances = 1	When a parity error occurs within the packet buffer memory, this indication is asserted. The location of the error is determined by reading `Packet_Buffer_Parity_Error_Info`. This indication is reset by writing a one to this bit location.
free_buffer_list_parity_error	Mode = R/W Offset = 0.26 Instances = 1	When a parity error occurs within the free buffer list memory, this indication is asserted. The location of the error is determined by reading `Packet_Buffer_Parity_Error_Info`. This indication is reset by writing a one to this bit location.
free_descriptor_list_parity_error	Mode = R/W Offset = 0.27 Instances = 1	When a parity error occurs within the free descriptor list memory, this indication is asserted. The location of the error is determined by reading `Packet_Buffer_Parity_Error_Info`. This indication is reset by writing a one to this bit location.
free_buffer_list_empty	Mode = R/W Offset = 0.28 Instances = 1	If a packet is discarded due to an exhaustion of free buffers, this indication is asserted. This indication is reset by writing a one to this bit location.

Appendix A: Registers (continued)

Packet_Buffer_Ind (continued)

Table 210. Packet_Buffer_Ind Field Parameters (continued)

Field Name	Parameter	Description
free_descriptor_list_empty	Mode = R/W Offset = 0.29 Instances = 1	If a packet is discarded due to an exhaustion of free descriptors, this indication is asserted. This indication is reset by writing a one to this bit location.
queue_full	Mode = R/W Offset = 0.30 Instances = 1	If a packet is discarded due to full queue, this indication is asserted. The identity of the queue that caused this indication is determined by reading `Packet_Buffer_Queue_Full_Info`. This indication is reset by writing a one to this bit location.
rate_adaptation_fifo_overflow	Mode = R/W Offset = 0.31 Instances = 1	If a packet is discarded due to an overflow of the rate adaptation FIFO (used during multicast replication), this indication is asserted. This indication is reset by writing a one to this bit location.
packet_buffer_ind_mask[6:0]	Mode = RO Offset = 4.25 Instances = 1	This field provides the supervisor with the current settings of the indication mask. Bits asserted in this field prevent the indication bits listed above from becoming asserted. Bits in `packet_buffer_ind_mask[6:0]` are set and cleared via the fields `packet_buffer_ind_mask_set[6:0]` and `packet_buffer_ind_mask_clear[6:0]` below.
packet_buffer_ind_mask_set [6:0]	Mode = WO Offset = 8.25 Instances = 1	Writing ones to bits in this field causes the corresponding bits to become asserted in `packet_buffer_ind_mask[6:0]`.
packet_buffer_ind_mask_clear [6:0]	Mode = WO Offset = 12.25 Instances = 1	Writing ones to bits in this field causes the corresponding bits to become deasserted in `packet_buffer_ind_mask[6:0]`.
packet_buffer_int[6:0]	Mode = RO Offset = 16.25 Instances = 1	The unmasked indication bits that are also not masked by `packet_buffer_int_mask[6:0]` are revealed to the supervisor via this field.
packet_buffer_int_mask[6:0]	Mode = RO Offset = 20.25 Instances = 1	This field provides the supervisor with the current settings of the indication mask. Bits asserted in this field prevent the interrupt bits in `packet_buffer_int[6:0]` from becoming asserted. Bits in `packet_buffer_int_mask[6:0]` are set and cleared via the fields `packet_buffer_int_mask_set[6:0]` and `packet_buffer_int_mask_clear[6:0]` below.
packet_buffer_int_mask_set [6:0]	Mode = WO Offset = 24.25 Instances = 1	Writing ones to bits in this field causes the corresponding bits to become asserted in `packet_buffer_int_mask[6:0]`.
packet_buffer_int_mask_clear [6:0]	Mode = WO Offset = 28.25 Instances = 1	Writing ones to bits in this field causes the corresponding bits to become deasserted in `packet_buffer_int_mask[6:0]`.

Various indications to the supervisor regarding packet_buffer. These indications may be masked by `Packet_Buffer_Ind_Mask`.

Appendix A: Registers (continued)

Packet_Buffer_Mode

Description: Mode bits.

Table 211. Packet_Buffer_Mode Register Parameters

Parameter	Value
Base Address	0x000c_bc64
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

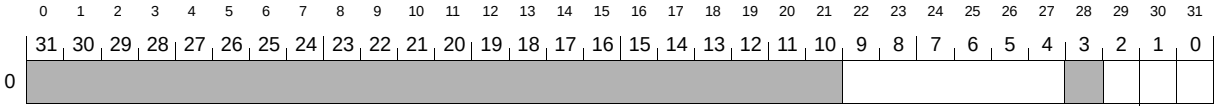


Figure 158. Packet_Buffer_Mode Register Diagram

Table 212. Packet_Buffer_Mode Field Parameters

Field Name	Parameter	Description
global_congestion_hysteresis[9:4]	Mode = R/W Offset = 22 Instances = 1	This field establishes the congestion hysteresis. The number of allocated buffers must fall below `global_congestion_threshold[14:8]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. This hysteresis value is shared by all congestion thresholds.
hol_mode	Mode = R/W Offset = 0.29 Instances = 1	When asserted, `packet_buffer` utilizes queue and buffer thresholds to manage and avoid head-of-line blocking scenarios in the shared resources. When this bit is asserted, the `Packet_Buffer_Queue_Management_Threshold`, `Packet_Buffer_Queue_Limit`, and `Packet_Buffer_Queue_Buffer_Limit` registers may be programmed. This bit is valid only in revision C and is reserved in versions B and B1.
bandwidth_provisioning_en	Mode = R/W Offset = 30 Instances = 1	This bit is asserted to enable the bandwidth provisioning (traffic shaping) algorithms. When this mode is enabled, a queue must be nonempty **and** below its bandwidth limit prior to transmission. When this mode is disabled, a queue need only be nonempty to arbitrate for transmission.

Various basic mode bits related to `packet_buffer`.

Appendix A: Registers (continued)

Packet_Buffer_Packet_Drop_Count

Description: Provides an accounting of various packet drop counts.

Table 213. Packet_Buffer_Packet_Drop_Count Register Parameters

Parameter	Value
Base Address	0x000c_bb00
Register Size (revisions B and B1)	52
Register Size (revision C)	56
Register Instances	1
Register Spacing	NA
Record Size (revisions B and B1)	52
Record Size (revision C)	56
Register Instances	1
Register Spacing	NA

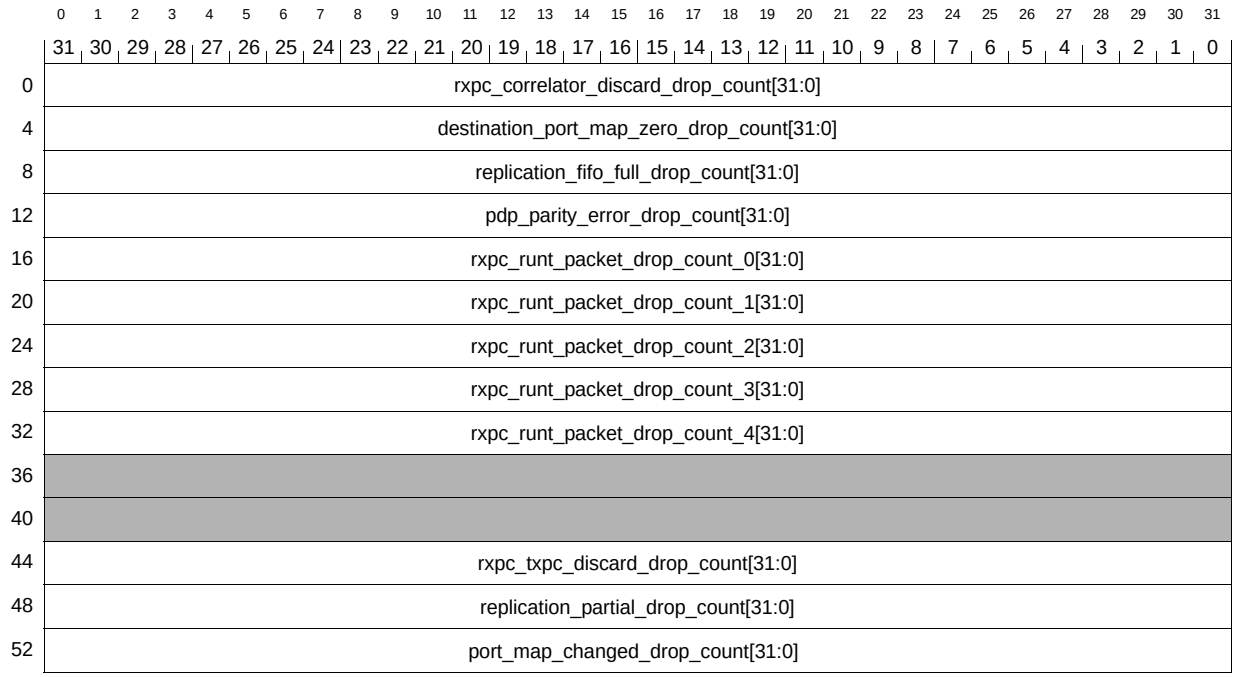


Figure 159. Packet_Buffer_Packet_Drop_Count Register Diagram

Appendix A: Registers (continued)

Packet_Buffer_Packet_Drop_Count (continued)

Table 214. Packet_Buffer_Drop_Count Field Parameters

Field Name	Parameter	Description
rxpc_correlator_discard_drop_count[31:0]	Mode = R/W Offset = 0.0 Instances = 1	The number of packet dropped because of the correlator function in the rxpc module.
destination_port_map_zero_drop_count[31:0]	Mode = R/W Offset = 4.0 Instances = 1	The number of packets dropped because it had a null (all zeros) destination map.
replication_fifo_full_drop_count[31:0]	Mode = R/W Offset = 8.0 Instances = 1	The number of packets dropped due to a full replication FIFO.
pdp_parity_error_drop_count[31:0]	Mode = R/W Offset = 12.0 Instances = 1	The number of packets dropped due to a parity error in the pdp module.
rxpc_runt_packet_drop_count_0[31:0]	Mode = R/W Offset = 16.0 Instances = 1	The number of runt packets dropped for `packet_processor 0`.
rxpc_runt_packet_drop_count_1[31:0]	Mode = R/W Offset = 20.0 Instances = 1	The number of runt packets dropped for `packet_processor 1`.
rxpc_runt_packet_drop_count_2[31:0]	Mode = R/W Offset = 24.0 Instances = 1	The number of runt packets dropped for `packet_processor 2`.
rxpc_runt_packet_drop_count_3[31:0]	Mode = R/W Offset = 28.0 Instances = 1	The number of runt packets dropped for `packet_processor 3`.
rxpc_runt_packet_drop_count_4[31:0]	Mode = R/W Offset = 32.0 Instances = 1	The number of runt packets dropped for `packet_processor 4`.
rxpc_txpc_discard_drop_count[31:0]	Mode = R/W Offset = 44.0 Instances = 1	The number of packets discarded due to a full packet buffer.
replication_partial_drop_count[31:0]	Mode = R/W Offset = 48.0 Instances = 1	The number of multicast packets partially enqueued due to full queues.
port_map_changed_drop_count[31:0]	Mode = R/W Offset =52.0 Instances = 1	The number of multicast packets causing modified port maps due to full queues. This field is valid only in revision C and is reserved in revisions B and B1.

Various basic mode bits related to `packet_buffer`.

Appendix A: Registers (continued)

Packet_Buffer_Parity_Error_Info

Description: Provides the location of the occurrence of a parity error.

Table 215. Packet_Buffer_Parity_Error_Info Register Parameters

Parameter	Value
Base Address	0x000c_bb78
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

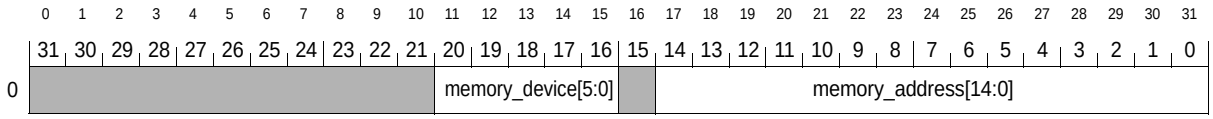


Figure 160. Packet_Buffer_Parity_Error_Info Register Diagram

Table 216. Packet_Buffer_Parity_Error_Info Field Parameters

Field Name	Parameter	Description
memory_device[5:0]	Mode = R/W Offset = 0.11 Instances = 1	The device that experienced the parity error.
memory_addr[14:0]	Mode = R/W Offset = 0.17 Instances = 1	The address of the parity error.

Upon the first occurrence of a parity error, this register stores and presents a value that identifies which device experienced the error and the address within the device that was being accessed at the time of the error. Further parity errors do not cause changes to the value held by this register. In order to reprime this register for the capture of a subsequent error, all `packet_buffer`-related parity error indication bits in `Packet_Buffer_Ind` must first be reset.

Appendix A: Registers (continued)

Packet_Buffer_Port_Speed

Description: Provides the port speed information to the traffic shapers.

Table 217. Packet_Buffer_Port_Speed Register Parameters

Parameter	Value
Base Address	0x000c_bd00
Register Size	192
Register Instances	1
Register Spacing	NA
Record Size	192
Record Instances	1
Record Spacing	NA

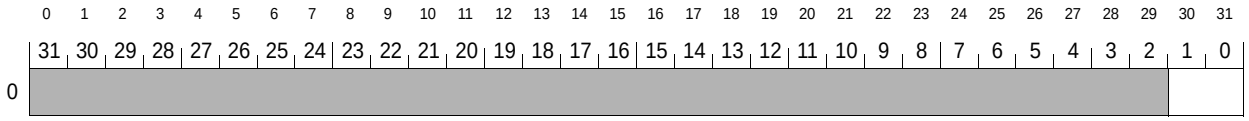


Figure 161. Packet_Buffer_Port_Speed Register Diagram

Table 218. Packet_Buffer_Port_Speed Field Parameters

Field Name	Parameter	Description
port_speed_{20..47}[1:0]	Mode = R/W Offset = 0.30 Instances = 48 Spacing = 4.0	The speed of the corresponding Ethernet MAC. 00_2 = 10 Mbits/s 01_2 = 100 Mbits/s 10_2 = 1 Gbit/s 11_2 = reserved

This register provides per-port interface speed information to the traffic shapers. This register does not actually affect the speed of the Ethernet ports; it merely informs the packet buffer of the speed at which each port is operating.

PORT NUMBERING SCHEME (TABLE INDEXING)

PORT	SU7	SU6		SU1	SU0	res	res	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	57	56		51	50	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 162. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Priority_Table

Description: Maps the 4-bit packet priority value to one of eight transmit queues.

Table 219. Packet_Buffer_Priority_Table Register Parameters

Parameter	Value
Base Address	0x000c_bc00
Register Size	64
Register Instances	1
Register Spacing	NA
Record Size	1
Record Instances	16
Record Spacing	4

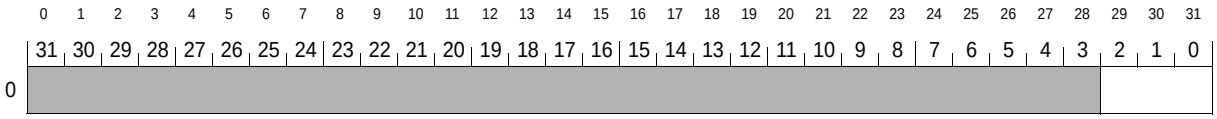


Figure 163. Packet_Buffer_Priority_Table Register Diagram

Table 220. Packet_Buffer_Priority_Table Field Parameters

Field Name	Parameter	Description
storage_priority_{0..15}[2:0]	Mode = R/W Offset = 0.29 Instances = 1	Storage priority value.

The 16 levels of priority utilized during ingress packet processing are used to select one of eight transmit queues associated with each transmit port. This table is used to map between various priority levels and queues. This table is addressed by the packet's priority level and returns the 3-bit queue selection value: `storage_priority[2:0]`.

Appendix A: Registers (continued)

Packet_Buffer_Queue_Buffer_Limit (Revision C Only)

Description: Global queue entry limit (in buffers) imposed when queue memory is congested.

Table 221. Packet_Buffer_Queue_Buffer_Limit Register Parameters

Parameter	Value
Base Address	0x000C_E200
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

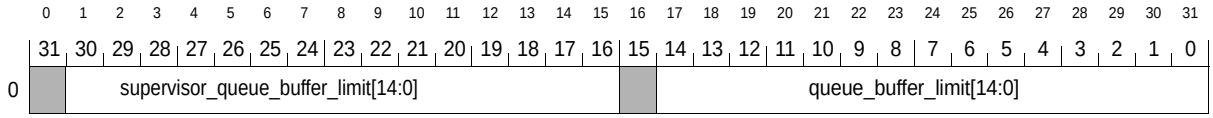


Figure 164. Packet_Buffer_Queue_Buffer_Limit Register Diagram

Table 222. Packet_Buffer_Queue_Buffer_Limit Field Parameters

Field Name	Parameters	Description
supervisor_queue_buffer_limit[14:0]	Mode = R/W Offset = 0.1 Instances = 1	Supervisor queue limit.
queue_buffer_limit[14:0]	Mode = R/W Offset = 0.17 Instances = 1	Queue limit.

This register defines the number of buffers (128 bytes) that each queue is allowed to use during times of packet buffer congestion. If the buffer memory is congested and the number of buffers used for any supervisor queue exceeds `supervisor_queue_buffer_limit[14:0]` or the number of queued buffers for any other queue exceeds `queue_buffer_limit[14:0]`, then that queue is disabled from receiving further entries. If a queue is in excess of this limit when the packet buffer becomes congested, its excess queue entries are not discarded; only new entries are inhibited from being enqueued.

This register is valid in revision C only, and the hol_mode bit in the Packet_Buffer_Mode register must be asserted to use this register.

Appendix A: Registers (continued)

Packet_Buffer_Queue_Buffers_{0..3} (Revision C Only)

Description: The number of buffers associated with each queue.

Table 223. Packet_Buffer_Queue_Buffers_{0..3} Register Parameters

Parameter	Value
Base Address	0x000C_DA00
Register Size	424
Register Instances	4
Register Spacing	512
Record Size	4
Record Instances	106
Record Spacing	4

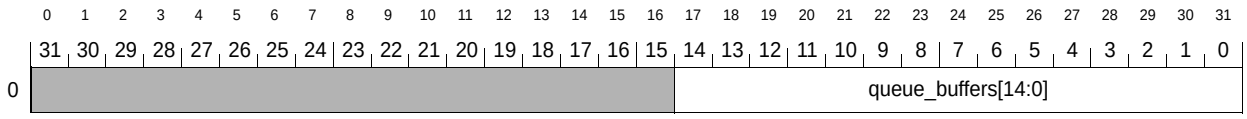


Figure 165. Packet_Buffer_Queue_Buffers_{0..3} Register Diagram

Table 224. Packet_Buffer_Queue_Buffers_{0..3} Field Parameters

Field Name	Parameter	Description
queue_buffers[14:0]	Mode = R/W Offset = 0.17 Instances = 1	The number of buffers associated with the corresponding queue.

This register provides the supervisor with the number of buffers that are currently associated with each of the queues. The queues are grouped according to the following table.

Appendix A: Registers (continued)

Packet_Buffer_Queue_Buffers_{0..3} (Revision C Only) (continued)

Table 225. Queue Groups

	Group 0	Group 1	Group 2	Group 3
Queues	0—7	8—15	16—23	24—31
	32—39	40—47	48—55	56—63
	64—71	72—79	80—87	88—95
	96—103	104—111	112—119	120—127
	128—135	136—143	144—151	152—159
	160—167	168—175	176—183	184—191
	192—199	200—207	208—215	216—223
	224—231	232—239	240—247	248—255
	256—263	264—271	272—279	280—287
	288—295	296—303	304—311	312—319
	320—327	328—335	336—343	344—351
	352—359	360—367	368—375	376—383
	384—391	384—391	392—399	392—399
	400	402	404	406
	401	403	405	407

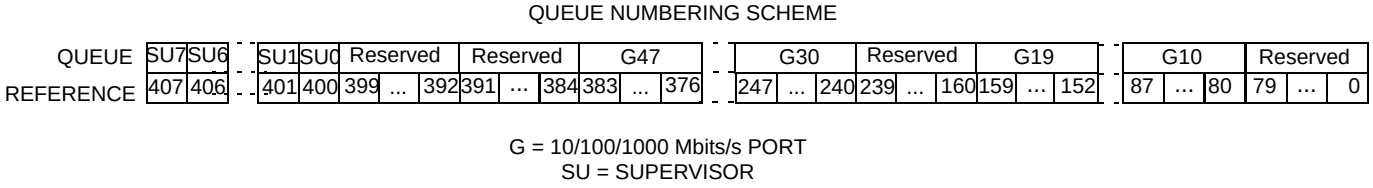


Figure 166. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Queue_Depth_{0..3} (Revision C Only)

Description: The number of entries in each queue.

Table 226. Packet_Buffer_Queue_Depth_{0..3} Register Parameters

Parameter	Value
Base Address	0x000C_D000
Register Size	428
Register Instances	4
Register Spacing	512
Record Size	4
Record Instances	107
Record Spacing	4

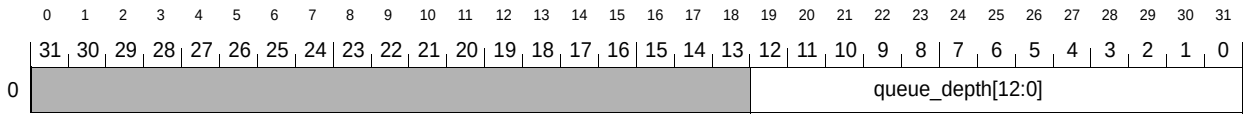


Figure 167. Packet_Buffer_Queue_Depth_{0..3} Register Diagram

Table 227. Packet_Buffer_Queue_Depth_{0..3} Field Parameters

Field Name	Parameter	Description
queue_depth[12:0]	Mode = R/W Offset = 0.19 Instances = 1	The depth of the corresponding queue in packets.

This register provides the supervisor with the number of entries that are currently in each of the queues. The queues are grouped according to the following table.

Appendix A: Registers (continued)

Packet_Buffer_Queue_Depth_{0..3} (Revision C Only) (continued)

Table 228. Queue Groups

	Group 0	Group 1	Group 2	Group 3
Queues	0—7	8—15	16—23	24—31
	32—39	40—47	48—55	56—63
	64—71	72—79	80—87	88—95
	96—103	104—111	112—119	120—127
	128—135	136—143	144—151	152—159
	160—167	168—175	176—183	184—191
	192—199	200—207	208—215	216—223
	224—231	232—239	240—247	248—255
	256—263	264—271	272—279	280—287
	288—295	296—303	304—311	312—319
	320—327	328—335	336—343	344—351
	352—359	360—367	368—375	376—383
	384—391	384—391	392—399	392—399
	400	402	404	406
	401	403	405	407
	Discard	Discard	Discard	Discard

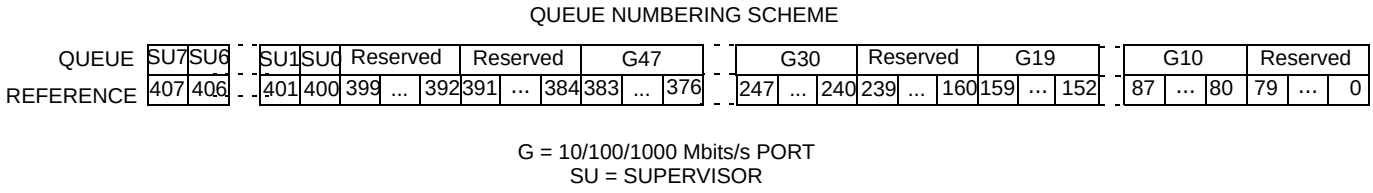


Figure 168. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Queue_En

Description: Per-queue enable.

Table 229. Packet_Buffer_Queue_En Register Parameters

Parameter	Value
Base Address	0x000c_a000
Register Size	1632
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	408
Record Spacing	4

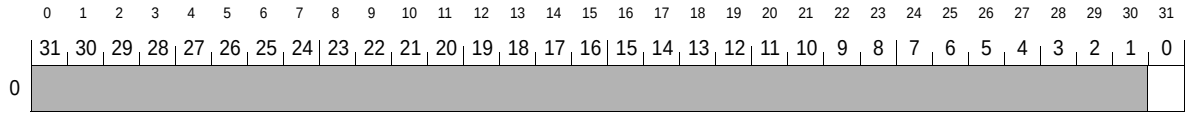


Figure 169. Packet_Buffer_Queue_En Register Diagram

Table 230. Packet_Buffer_Queue_En Field Parameters

Field Name	Parameter	Description
queue_weight[0]	Mode = R/W Offset = 0.31 Instances = 1	The enable for the corresponding queue.

Queues may be individually enabled via this register. A disabled queue does not arbitrate for transmission. However, a disabled queue also does not prevent further enqueuing. Hence, a disabled queue that remains a destination will quickly fill up.

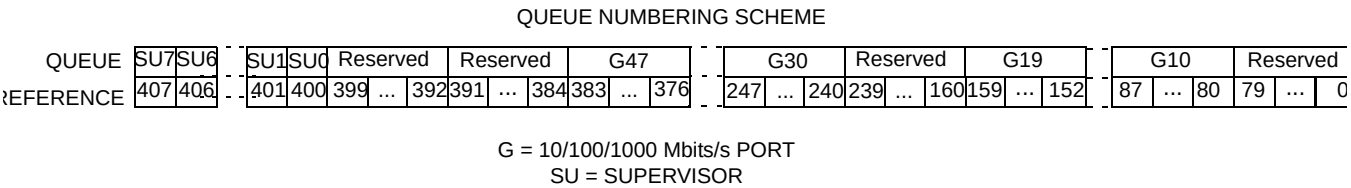


Figure 170. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Queue_Limit (Revision C Only)

Description: Global queue entry in buffer limit imposed when queue memory is congested.

Table 231. Packet_Buffer_Queue_Limit Register Parameters

Parameter	Value
Base Address	0x000C_E204
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

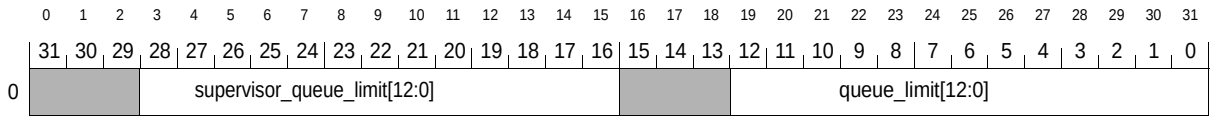


Figure 171. Packet_Buffer_Queue_Limit Register Diagram

Table 232. Packet_Buffer_Queue_Limit Field Parameters

Field Name	Parameters	Description
supervisor_queue_limit[12:0]	Mode = R/W Offset = 0.1 Instances = 1	Supervisor queue limit.
queue_limit[12:0]	Mode = R/W Offset = 0.17 Instances = 1	Queue limit.

This register defines the number of queue entries in packets that each queue is allowed to have during times of queue memory congestion. If the queue memory is congested and the number of queue entries for any supervisor queue exceeds the `supervisor_queue_limit[12:0]` or for any other queue exceeds the `queue_limit[12:0]` value, then that queue is disabled from receiving further entries. If a queue is in excess of its limit when the queue memory becomes congested, its excess queue entries are not discarded; only new entries are inhibited from being enqueued.

This register is valid in revision C only, and the hol_mode bit in the packet_buffer_mode register must be asserted to use this register.

Appendix A: Registers (continued)

Packet_Buffer_Queue_Management_Thresholds (Revision C Only)

Description: Global queue congestion thresholds.

Table 233. Packet_Buffer_Queue_Management_Thresholds Register Parameters

Parameter	Value
Base Address	0x000C_E208
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

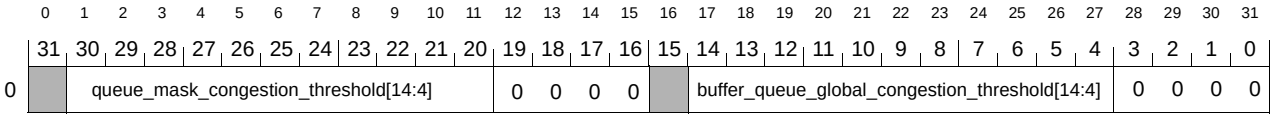


Figure 172. Packet_Buffer_Queue_Management_Thresholds Register Diagram

Table 234. Packet_Buffer_Management_Thresholds Field Parameters

Field Name	Parameters	Description
queue_mask_congestion_ threshold[14:4]	Mode = R/W Offset = 0.1 Instances = 1	When the number of queue entries allocated equals or exceeds this value, the queue memory is considered congested. The number of queue entries allocated must drop below `queue_mask_congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. The results of this congestion test are used in determining which queues to disable for further enqueues. If a queue memory exceeds this threshold, all queues associated with the memory that also exceed `packet_buffer_queue_limit` are to be masked from further entries.
buffer_queue_global_congestion_ threshold[14:4]	Mode = R/W Offset = 0.17 Instances = 1	When the total number of queued buffers allocated equals or exceeds this value, the packet buffer is considered congested. The number of queued buffers allocated must drop below `buffer_queue_global_congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]` for the state to change from congested to not congested. The results of this congestion test are used in determining which queues to disable for further enqueues. If a queue memory exceeds this threshold, all queues associated with the memory that also exceed `packet_buffer_queue_buffer_limit` are to be masked from further entries.

This register is valid in revision C only, and the hol_mode bit in the packet_buffer_mode register must be asserted to use this register.

Appendix A: Registers (continued)

Packet_Buffer_Queue_Status

Description: Provides a queue status overview.

Table 235. Packet_Buffer_Queue_Status Register Parameters

Parameter	Value
Base Address	0x000c_ba00
Register Size	200
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	50
Record Spacing	4

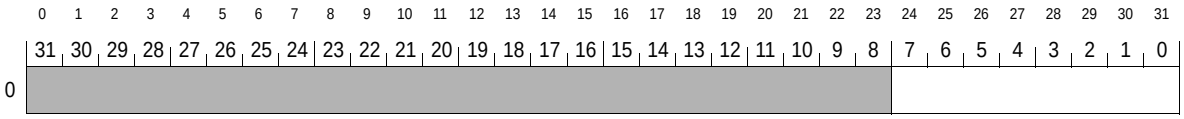


Figure 173. Packet_Buffer_Queue_Status Register Diagram

Table 236. Packet_Buffer_Queue_Status Field Parameters

Field Name	Parameter	Description
queue_not_empty[7:0]	Mode = RO Offset = 0.24 Instances = 1	Bits are asserted to indicate that the corresponding queue is not empty.

Each active record of this register corresponds to one of the system's 30 transmit ports (28 Ethernet ports and 2 supervisor ports). Within `queue_not_empty[7:0]`, the left-most bit corresponds to queue priority 7 (highest priority), while the right-most bit corresponds to queue priority 0 (lowest priority). When one of queues is not empty, its corresponding bit is asserted. When the queue is empty, its corresponding status bit is deasserted.

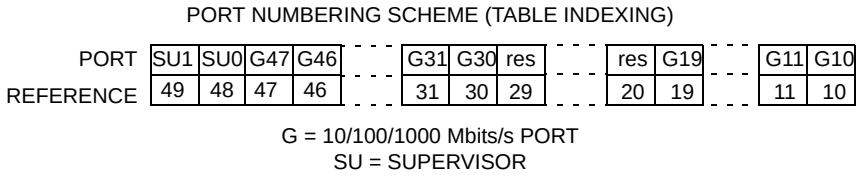


Figure 174. Port Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Scrub

Description: Initiates and directs the scrubbing of lost buffers.

Table 237. Packet_Buffer_Scrub Register Parameters

Parameter	Value
Base Address	0x000c_bb80
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

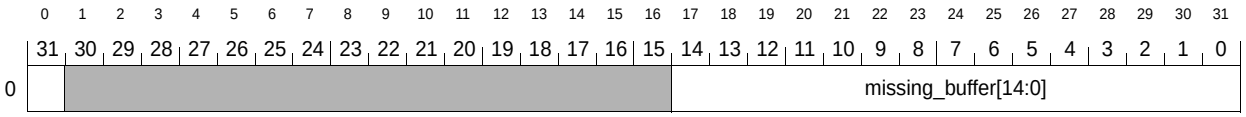


Figure 175. Packet_Buffer_Scrub Register Diagram

Table 238. Packet_Buffer_Scrub Field Parameters

Field Name	Parameter	Description
scrubbing	Mode = RO Offset = 0.0 Instances = 1	This bit is asserted while the scrubbing function is in progress. No writes are permitted to this register while `scrubbing` is asserted.
missing_buffer[14:0]	Mode = WO Offset = 0.17 Instances = 1	The buffer number of the buffer to be searched for.

The supervisor writes a buffer number to this register to perform a scrubbing operation. When the buffer is found, it is placed back on the free list.

Appendix A: Registers (continued)

Packet_Buffer_Shaper_Accumulator_Even

Description: Up/down accumulators used for traffic shaping.

Table 239. Packet_Buffer_Shaper_Accumulator_Even Register Parameters

Parameter	Value
Base Address	0x000c_a800
Register Size	816
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	204
Record Spacing	4

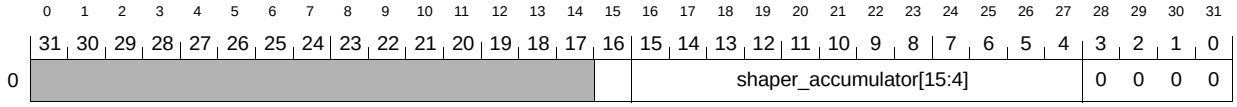


Figure 176. Packet_Buffer_Shaper_Accumulator_Even

Table 240. Packet_Buffer_Shaper_Accumulator_Even Field Parameters

Field Name	Parameter	Description
shaper_accumulator_sign	Mode = R/W Offset = 0.15 Instances = 1	The sign of the two's complement `shaper_accumulator[15:4]` value. When asserted, this signal indicates that `shaper_accumulator[15:4]` is negative.
shaper_accumulator[15:4]	Mode = R/W Offset = 0.16 Instances = 1	The current value of a traffic shaper accumulator. This counter does not count below its negative minimum or above its positive maximum.

The shaper accumulators are used to determine when a particular queue may have one of its packet's scheduled for retrieval and transmission. A queue is considered eligible for scheduling when its corresponding `shaper_accumulator[15:4]` value is greater than zero.

There is one `shaper_accumulator[15:4]` value per queue.

The byte count of a transmitted packet is subtracted from its queue's associated `shaper_accumulator[15:4]` value. Shaper credits are periodically added to `shaper_accumulator[15:4]`.

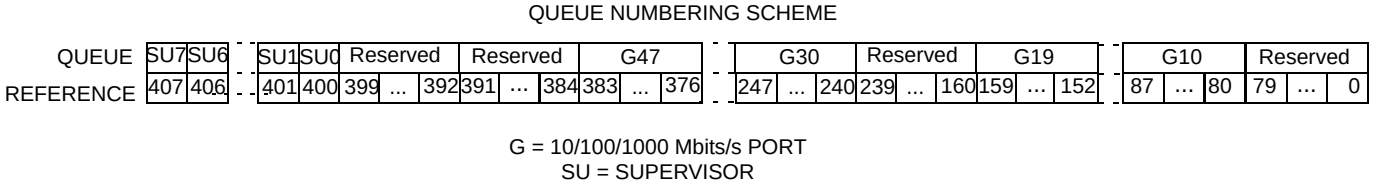


Figure 177. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Shaper_Accumulator_Odd

Description: Up/down accumulators used for traffic shaping.

Table 241. Packet_Buffer_Shaper_Accumulator_Odd Register Parameters

Parameter	Value
Base Address	0x000c_ac00
Register Size	816
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	204
Record Spacing	4

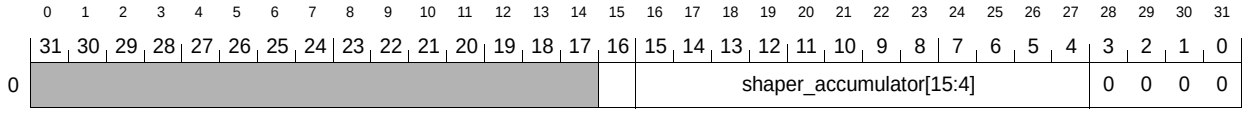


Figure 178. Packet_Buffer_Shaper_Accumulator_Odd Register Diagram

Table 242. Packet_Buffer_Shaper_Accumulator_Odd Field Parameters

Field Name	Parameter	Description
shaper_accumulator_sign	Mode = R/W Offset = 0.15 Instances = 1	The sign of the two's complement `shaper_accumulator[15:4]` value. When asserted, this signal indicates that `shaper_accumulator[15:4]` is negative.
shaper_accumulator[15:4]	Mode = R/W Offset = 0.16 Instances = 1	The current value of a traffic shaper accumulator. This counter does not count below its negative minimum or above its positive maximum.

The shaper accumulators are used to determine when a particular queue may have one of its packet's scheduled for retrieval and transmission. A queue is considered eligible for scheduling when its corresponding `shaper_accumulator[15:4]` value is greater than zero.

There is one `shaper_accumulator[15:4]` value per queue.

The byte count of a transmitted packet is subtracted from its queue's associated `shaper_accumulator[15:4]` value. Shaper credits are periodically added to `shaper_accumulator[15:4]`.

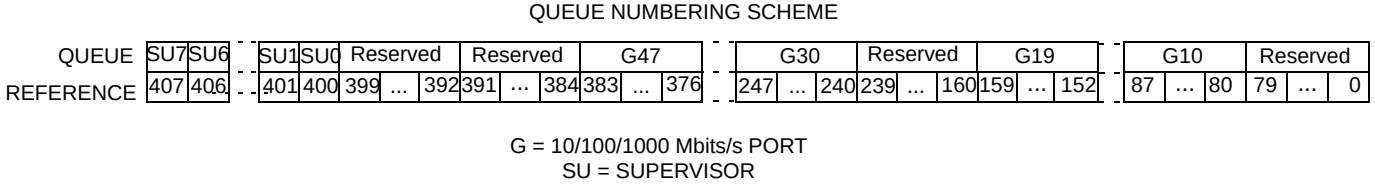


Figure 179. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Shaper_Delta_Even

Description: These values determine a shaper's target byte rate.

Table 243. Packet_Buffer_Shaper_Delta_Even Register Parameters

Parameter	Value
Base Address	0x000c_b000
Register Size	816
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	204
Record Spacing	4

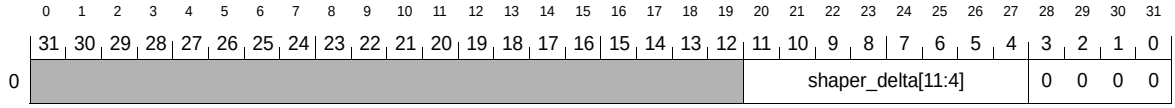


Figure 180. Packet_Buffer_Shaper_Delta_Even Register Diagram

Table 244. Packet_Buffer_Shaper_Delta_Even Field Parameters

Field Name	Parameter	Description
shaper_delta[11:4]	Mode = R/W Offset = 0.20 Instances = 1	The credit that is periodically added to an accumulator.

On a periodic basis, the value of `shaper_delta[11:4]` is added to its corresponding `Packet_Buffer_Shaper_Accumulator` register field. The period of these applications of credit is derived from the system clock and is dependent on the bit rate of the associated interface. If a period of p is assumed for a 1 Gbit/s Ethernet port, then 10p is used for 100 Mbits/s and 100 p for 10 Mbits/s. The nominal value for p is 12.8 µs, or 78,125 credits per second.

When updated 78,125 times per second for a 1 Gbit/s port, a `shaper_delta[11:0]` value of 16 (shaper_delta[11:4] = 0x1) results in a transmit byte rate of 1.25 Mbytes/s, or 10 Mbits/s. This is 1% of 1 Gbit/s. A `shaper_delta[11:0]` value of 1,600 results in a transmit byte rate of 125 Mbytes/s, or 1 Gbit/s. This is, of course, 100% of the interface's rated speed.

For lower-speed interfaces, the credit application rate is reduced proportionately. Therefore, the same values of 16 and 1,600 can be used in `shaper_delta[11:0]` as the minimum and maximum data rates.

Port speed is indicated by values set in `Packet_Buffer_Port_Speed` by the supervisor.

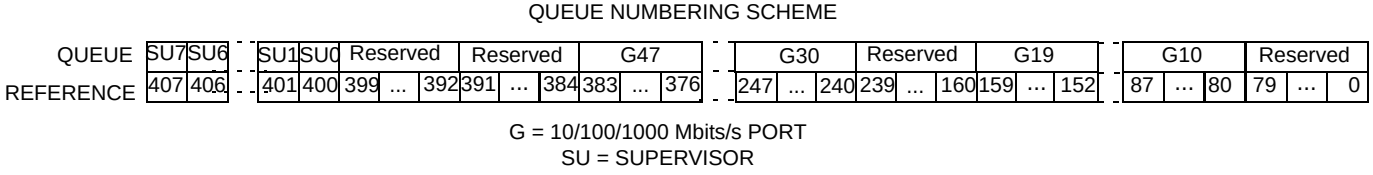


Figure 181. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Shaper_Delta_Odd

Description: These values determine a shaper's target byte rate.

Table 245. Packet_Buffer_Shaper_Delta_Odd Register Parameters

Parameter	Value
Base Address	0x000c_b400
Register Size	816
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	204
Record Spacing	4

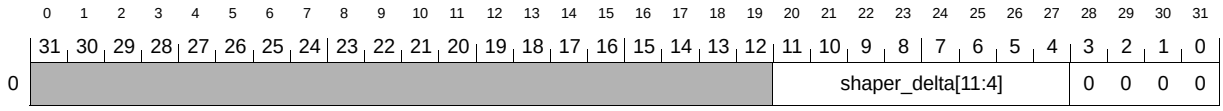


Figure 182. Packet_Buffer_Shaper_Delta_Odd Register Diagram

Table 246. Packet_Buffer_Shaper_Delta_Odd Field Parameters

Field Name	Parameter	Description
shaper_delta[11:4]	Mode = R/W Offset = 0.20 Instances = 1	The credit that is periodically added to an accumulator.

On a periodic basis, the value of `shaper_delta[11:4]` is added to its corresponding `Packet_Buffer_Shaper_Accumulator` register field. The period of these applications of credit is derived from the system clock and is dependent on the bit rate of the associated interface. If a period of p is assumed for a 1 Gbit/s Ethernet port, then 10p is used for 100 Mbits/s and 100p for 10 Mbits/s. The nominal value for p is 12.8 microsecond; or 78,125 credits per second.

When updated 78,125 times per second for a 1 Gbit/s port, a `shaper_delta[11:0]` value of 16 (shaper_delta[11:4] = 0x1) results in a transmit byte rate of 1.25 Mbytes/s, or 10 Mbits/s. This is 1% of 1 Gbit/s. A `shaper_delta[11:0]` value of 1,600 results in a transmit byte rate of 125 Mbytes/s, or 1 Gbit/s. This is, of course, 100% of the interface's rated speed.

For lower-speed interfaces, the credit application rate is reduced proportionately. Therefore, the same values of 16 and 1,600 can be used in `shaper_delta[11:0]` as the minimum and maximum data rates.

Port speed is indicated by values set in `Packet_Buffer_Port_Speed` by the supervisor.

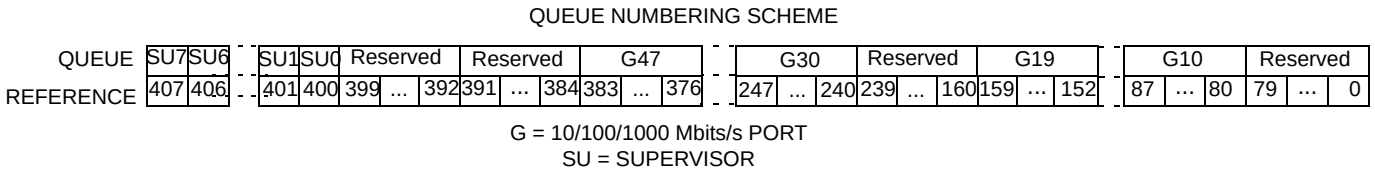


Figure 183. Queue Numbering Scheme

Appendix A: Registers (continued)

Packet_Buffer_Shaper_Limit

Description: An upper limit for shaper accumulators.

Table 247. Packet_Buffer_Shaper_Limit Register Parameters

Parameter	Value
Base Address	0x000c_bc68
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

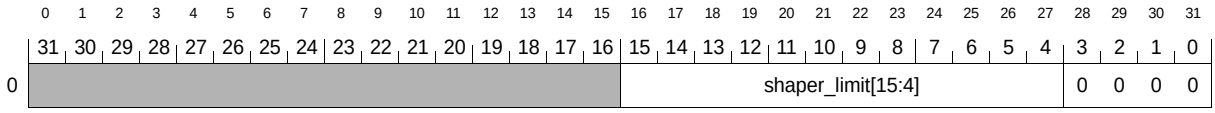


Figure 184. Packet_Buffer_Shaper_Limit Register Diagram

Table 248. Packet_Buffer_Shaper_Limit Field Parameters

Field Name	Parameter	Description
shaper_limit[15:4]	Mode = R/W Offset = 0.16 Instances = 1	Establishes an upper limit for each shaper accumulator.

This register is used to establish an upper limit for `Packet_Buffer_Shaper_Accumulator.shaper_accmulator[15:4]`. If a `shaper_accmulator[15:4]` value ever equals or exceeds its corresponding `shaper_limit[15:4]` value, then no further increases in `shaper_accumulator[15:4]` occur. Under no circumstances is `shaper_accumulator[15:0]` permitted to exceed FFF0$_{16}$.

Appendix A: Registers (continued)

Policer_Accumulator_Table_{0..4}

Description: This table is used to accumulate packet data rates.

Table 249. Policer_Accumulator_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_d400
Register Size	320
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	80
Record Spacing	4

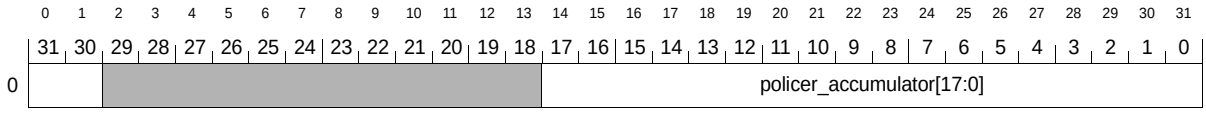


Figure 185. Policer_Accumulator_Table_{0..4} Register Diagram

Table 250. Policer_Accumulator_Table_{0..4} Field Parameters

Field Name	Parameter	Description
port_speed[1:0]	Mode = R/W Offset = 0.0 Instances = 4	The speed of the associated port. This field is defined as follows: 00_2 = 1 Gbit/s 01_2 = 100 Mbits/s 10_2 = 10 Mbits/s 11_2 = reserved
policer_accumulator[17:0]	Mode = R/W Offset = 0.14 Instances = 4 Spacing = 4.0	The policer accumulator. There are eight policers per port.

`Rx_Packet_Header.packet_length[13:0]` is added to the `policer_accumulator[17:0]` field that corresponds to the packet's policer for each received packet. On regular intervals, `policer_delta[7:0]` is subtracted from `policer_accumulator[17:0]`. If `policer_accumulator[17:0]` exceeds `policer_limit[17:11]`, then the traffic is considered out of profile.

Note: Policer_Accumulator_Table_{0} and Policer_Accumulator_Table_{2} are reserved and are not used.

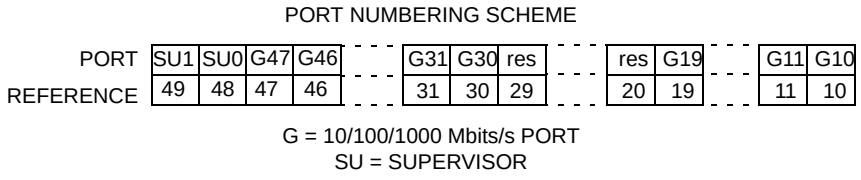


Figure 186. Port Numbering Scheme

Appendix A: Registers (continued)

Policer_Delta_Table_{0..4}

Description: This table is used to specify the per-flow leak rates for the channelized packet processors.

Table 251. Policer_Delta_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_d600
Register Size	320
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	80
Record Spacing	4

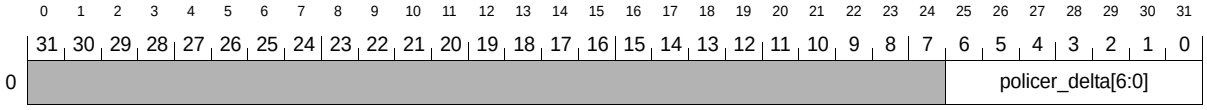


Figure 187. Policer_Delta_Table_{0..4} Register Diagram

Table 252. Policer_Delta_Table_{0..4} Field Parameters

Field Name	Parameter	Description
policer_delta[6:0]	Mode = R/W Offset = 0.25 Instances = 1	The policer delta.

The product of the `policer_delta[6:0]` value in this table and 16 are subtracted from their corresponding entries in the `Policer_Accumulator_Table_{0..4}` at a regular interval. The interval is derived from the system clock and varies from policer to policer depending on the data rate of the associated Ethernet receive port.

The nominal service interval for a 1 Gbit/s port is once every 12.8 µs or 78,125 times per second. In addition, the service interval for a 100 Mbits/s port is 1/10 that of a 1 Gbit/s port; and for a 10 Mbits/s port, 1/100 that of a 1 Gbit/s port. This scaling of service intervals provides the necessary data rate adaptation.

Larger `policer_delta[6:0]` values translate into higher allowed data rates on the corresponding policer.

When subtracted from the `Policer_Accumulator_Table_{0..4},` a product of 16 (`policer_delta[6:0]` = 0x1) results in a service interval of 1.25 Mbytes/s, or 10 Mbits/s. This is 1% of 1 Gbit/s. A product of 1,600 (policer_delta[6:0] = 0x64) results in a service interval of 125 Mbytes/s, or 1 Gbit/s. This is, of course, 100% of the interface's rated speed.

For lower-speed interfaces, the interval is reduced proportionately. Therefore, the same values of 1 and 100 can be used in `policer_delta[6:0]` for the minimum and maximum intervals.

Note: Policer_Delta_Table_{0} and Policer_Delta_Table_{2} are reserved and are not used.

Appendix A: Registers (continued)

Policer_Delta_Table_{0..4} (continued)

PORT NUMBERING SCHEME

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 188. Port Numbering Scheme

Appendix A: Registers (continued)

Policer_Flow_Id_Table_{0..4}

Description: This table is used to convert a 3-tuple to a single flow identifier.

Table 253. Policer_Flow_Id_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_c000
Register Size	4096
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	1024
Record Spacing	4

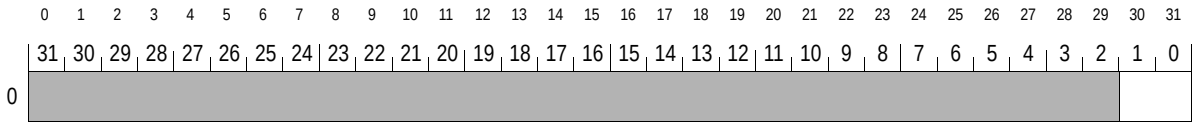


Figure 189. Policer_Flow_Id_Table_{0..4} Register Diagram

Table 254. Policer_Flow_Id_Table_{0..4} Field Parameters

Field Name	Parameter	Description
policer_flow_id[1:0]	Mode = R/W Offset = 0.30 Instances = 1	The policer flow identifier.

A received packet's source address and destination address are each reduced to 3-bit flow identifiers by their respective look-up processes. These two identifiers are concatenated with the packet's encoded 4-bit priority value to form a 10-bit address into this table. The following shows the form of this value:

flow_id_table_addr[9:0] = { dest_flow_id[2:0], src_flow_id[2:0], priority[3:0] };

The `dest_flow_id[2:0]` and `src_flow_id[2:0]` values are derived from either the packet's Layer 2 address look-up or its Layer 3 address look-up. The Layer 3 values are used if the packet contains a valid IPv4 or IPv6 header and `Policer_Mode.layer_2_flow_id_override_en` is not asserted. Otherwise, the packet's Layer 2 destination and source address values are used.

These tables are associated with the 1 gigabit Ethernet ports.

Note: Policer_Flow_Id_Table_{0} and Policer_Flow_Id_Table_{2} are reserved and are not used.

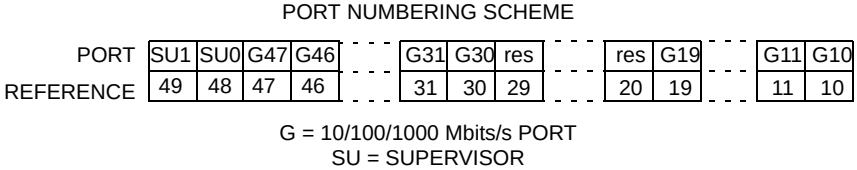


Figure 190. Port Numbering Scheme

Appendix A: Registers (continued)

Policer_Flow_Mode_Table_{0..4}

Description: This table per-flow mode settings.

Table 255. Policer_Flow_Mode_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_d800
Register Size	320
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	80
Record Spacing	4

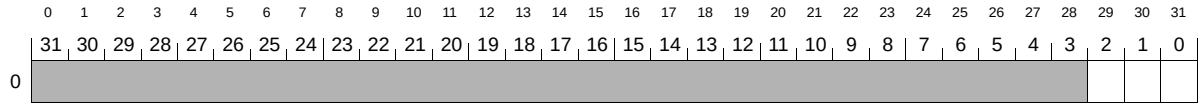


Figure 191. Policer_Flow_Mode_Table_{0..4} Register Diagram

Table 256. Policer_Flow_Mode_Table_{0..4} Field Parameters

Field Name	Parameter	Description
discard_out_of_profile_en	Mode = R/W Offset = 0.29 Instances = 80 Spacing = 4.0	This bit enables the discarding of packets that are out of profile. A packet is out of profile if its contribution to a flow's short-term average bandwidth causes that average to exceed the flow's user-defined limit.
discard_out_of_profile_when_ congested_en	Mode = R/W Offset = 0.30 Instances = 80 Spacing = 4.0	This bit enables the discarding of packets that are out of profile while the switch is congested. The switch is considered congested when the number of available packet buffer, packet queue or packet descriptor resources falls below user-defined thresholds.
demote_out_of_profile_en	Mode = R/W Offset = 0.31 Instances = 80 Spacing = 4.0	This bit enables the demotion of packets that are out of profile. When a packet is demoted, it is transmitted at a lower-priority level.

This register provides an assortment of flow-specific mode settings. For the channelized packet_processor modules (numbers 0 through 4), the 80 policer mode sets are allocated eight per channel. Policers 0 through 7 are allocated to channel 0, policers 8 through 15 to channel 1, etc.

Note: Policer_Flow_Mode_{0} and Policer_Flow_Mode_{2} are reserved and are not used.

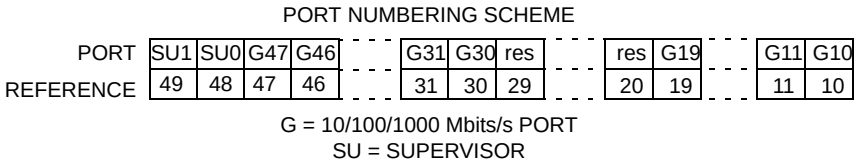


Figure 192. Port Numbering Scheme

Appendix A: Registers (continued)

Policer_Limit_Table_{0..4}

Description: Establishes the bandwidth limit for each flow.

Table 257. Policer_Limit_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0004_da00
Register Size	320
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	80
Record Spacing	4

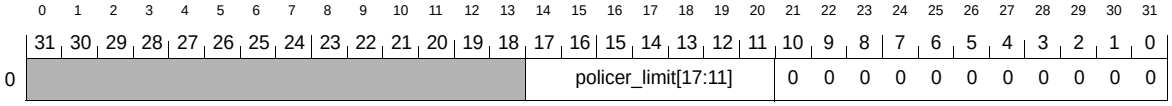


Figure 193. Policer_Limit_Table_{0..4} Register Diagram

Table 258. Policer_Limit_Table_{0..4} Field Parameters

Field Name	Parameter	Description
policer_limit[17:11]	Mode = R/W Offset = 0.14 Instances = 1	The per-flow policer limit.

If a `policer_accumulator[17:0]` value ever exceeds its corresponding `policer_limit[17:11]` value, then the flow is considered out of profile.

Note: Policer_Limit_Table_{0} and Policer_Limit_Table_{2} are reserved and are not used.

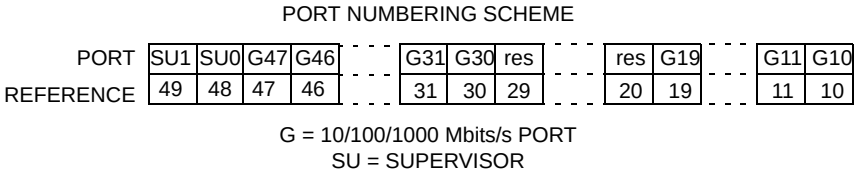


Figure 194. Port Numbering Scheme

Appendix A: Registers (continued)

Policer_Mode_{0..4}

Description: Mode bits for the policer function.

Table 259. Policer_Mode_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_dfe0
Register Size	4
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	1
Record Spacing	NA

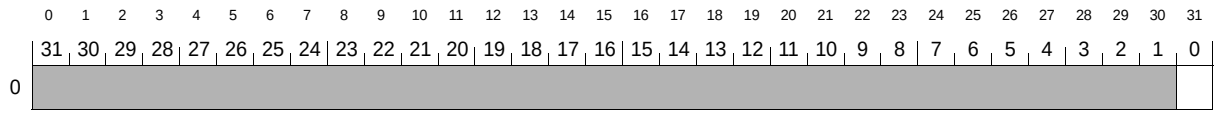


Figure 195. Policer_Mode_{0..4} Register Diagram

Table 260. Policer_Mode_{0..4} Field Parameters

Field Name	Parameter	Description
policer_en	Mode = R/W Offset = 0.31 Instances = 1 Reset = 0	Enables the policer function.

Each `packet_processor` module includes an aggregate policer function. Each aggregate consists of 80 policer functions. The mode bit in this register acts globally on a policer aggregate, not on individual policers.

Note: Policer_Mode_{0} and Policer_Mode_{2} are reserved and are not used.

PORT NUMBERING SCHEME

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 196. Port Numbering Scheme

Appendix A: Registers (continued)

Policer_Statistics_{0..4}

Description: Tracks policer behavior.

Table 261. Policer_Statistics_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_e000
Register Size	640
Register Instances	5
Register Spacing	32768
Record Size	8
Record Instances	80
Record Spacing	8

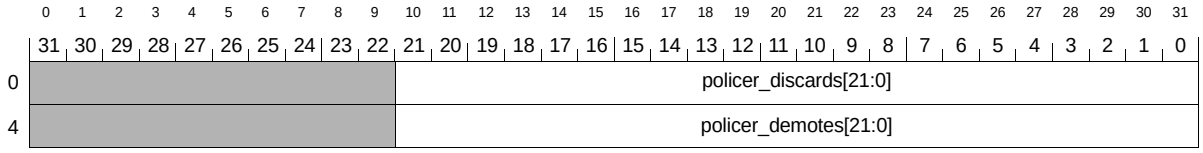


Figure 197. Policer_Statistics_{0..4} Register Diagram

Table 262. Policer_Statistics_{0..4} Field Parameters

Field Name	Parameter	Description
policer_discards[21:0]	Mode = R/W Offset = 0.10 Instances = 1	The count of the number of packets discarded per policer.
policer_demotes[21:0]	Mode = R/W Offset = 4.10 Instances = 1	The count of the number of packets demoted per policer.

These statistics counters increment by one for each packet that matches the associated criteria. These counters roll over to zero upon incrementing beyond its maximum value. It is incumbent upon the supervisor to sample these statistics often enough to avoid missing a counter rollover.

There is one pair of counters for each policer. For the multichannel `packet_processor` modules, the statistics are arranged such that the statistics type (demote vs. discard) forms the least-significant bit of the address. The policer number forms the next most-significant 3 bits, and the channel number forms the most-significant 4 bits of the statistics selection address.

Note: Policer_Statistics_{0} and Policer_Statistics_{2} are reserved and are not used.

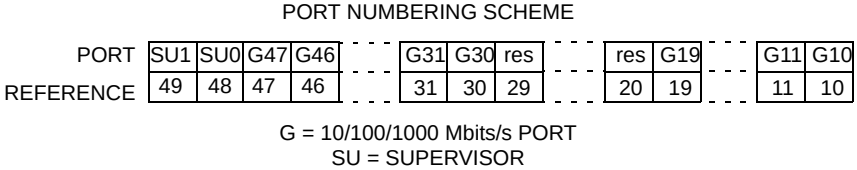


Figure 198. Port Numbering Scheme

Appendix A: Registers (continued)

Port_Mode_{0..4}

Description: Various port-oriented mode bits and fields for the multichannel packet processors.

Table 263. Port_Mode_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_df80
Register Size	40
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	10
Record Spacing	4

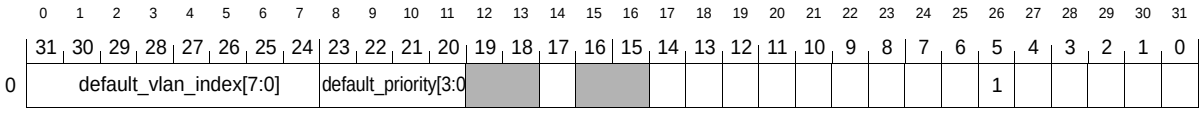


Figure 199. Port_Mode_{0..4} Register Diagram

Table 264. Port_Mode_{0..4} Field Parameters

Field Name	Parameter	Description
default_vlan_index_{}{0..9}[7:0]	Mode = R/W Offset = 0.0 Instances = 1 Reset = 0	The port's default VLAN index value.
default_priority_{}{0..9}[3:0]	Mode = R/W Offset = 0.8 Instances = 1 Reset = 0	The port's default priority value. If Layer 3 and Layer 2 priority information is not available from the received packet, this default priority value is used instead.
header_update_disable_{}{0..9}	Mode = R/W Offset = 0.14 Instances = 1 Reset = 0	This bit is asserted to disable the updating of a packet's Layer 2 header (VLAN tag, CoS) prior to transmission. Set this bit for an egress port if tagged or untagged ingress packets should be transmitted unchanged as tagged or untagged, respectively, regardless of an access or trunk-configured egress port. Also, if a port has been designated as a mirroring port, its `header_update_disable` bit should be asserted. Otherwise, this bit should be cleared.
bad_length_discard_en_{}{0..9}	Mode = R/W Offset = 0.17 Instances = 1 Reset = 0	If a received packet's length field does not match the length of the packet's data field, and this bit is asserted, then the packet is marked for discard.

Appendix A: Registers (continued)

Port_Mode_{0..4} (continued)

Table 264. Port_Mode_{0..4} Field Parameters (continued)

Field Name	Parameter	Description
unknown_encap_discard_en_{}{0..9}	Mode = R/W Offset = 0.18 Instances = 1 Reset = 0	If a received packet's encapsulation is not Ethernet V2 or SNAP (with or without VLAN tag), and this mode bit is asserted, then the packet is discarded.
unknown_ethertype_discard_en_{}{0..9}	Mode = R/W Offset = 0.19 Instances = 1 Reset = 0	If a received packet's Ethertype field is not IPv4, IPv6, ARP, or RARP, and this mode bit is asserted, then the packet is discarded.
unknown_ip_protocol_discard_en_{}{0..9}	Mode = R/W Offset = 0.20 Instances = 1 Reset = 0	If a received packet's IP protocol field is not ICMP, IGMP, TCP, or UDP, and this mode bit is asserted, then the packet is discarded.
ipv4_options_discard_en_{}{0..9}	Mode = R/W Offset = 0.21 Instances = 1 Reset = 0	If an IPv4 packet contains options and this mode bit is asserted, then the packet is discarded.
vlan_tagged_discard_en_{}{0..9}	Mode = R/W Offset = 0.22 Instances = 1 Reset = 0	If a received packet is tagged with a VLAN identifier, and this mode bit is asserted, then the packet is discarded.
vlan_untagged_discard_en_{}{0..9}	Mode = R/W Offset = 0.23 Instances = 1 Reset = 0	If a received packet is tagged with a VLAN identifier, and this mode bit is asserted, then the packet is discarded.
cfi_discard_en_{}{0..9}	Mode = R/W Offset = 0.24 Instances = 1 Reset = 0	If a received packet is VLAN tagged and its CFI bit is asserted, and this mode bit is asserted, then the packet is discarded.
layer_2_priority_override_en_{}{0..9}	Mode = R/W Offset = 0.25 Instances = 1 Reset = 0	This bit is asserted to force the use of Layer 2 priority information regardless of the presence of valid Layer 3 information. When this bit is deasserted, Layer 3 priority information is used if valid IPv4 or IPv6 headers are present.
reserved, set to one	Mode = R/W Offset = 0.26 Instances = 1 Reset = 0	This mode bit must be set to one for normal operation.
invalid_vlan_id_discard_en_{}{0..9}	Mode = R/W Offset = 0.27 Instances = 1 Reset = 0	If a received packet's VLAN ID value is not found in the VLAN index table and this bit is asserted, then the packet is discarded.

Appendix A: Registers (continued)

Port_Mode_{0..4} (continued)

Table 264. Port_Mode_{0..4} Field Parameters (continued)

Field Name	Parameter	Description
strip_matching_vlan_tag_en_{}{0..9}	Mode = R/W Offset = 0.28 Instances = 1 Reset = 0	If a transmit packet's VLAN index matches a trunk port's default VLAN index and this bit is asserted, the VLAN identifier is stripped from the packet (if there is one) prior to transmission.
trunk_port_{}{0..9}	Mode = R/W Offset = 0.29 Instances = 1 Reset = 0	This bit is asserted to indicate that a port is a VLAN trunk port.
update_cos_en_{}{0..9}	Mode = R/W Offset = 0.30 Instances = 1 Reset = 0	If this mode bit is asserted, then the Layer 2 CoS field is updated prior to transmission with the priority value computed during ingress processing. If this transmit port's `trunk_port` bits is not set, `update_cos_en` has no effect.
update_dscp_en_{}{0..9}	Mode = R/W Offset = 0.31 Instances = 1 Reset = 0	If this mode bit is asserted and the packet being transmitted is an IPv4 or IPv6 packet, then the priority value computed during ingress processing is decoded into a replacement DSCP field and placed in the transmit packet.

A collection of per-port mode bits.

This register description applied to the multichannel packet_processors (numbers 0 through 4) that service the 1 Gbit/s Ethernet ports. For each instance of this register, there are ten instances of its multifield record. Record zero corresponds to channel zero.

Note: Port_Mode_{0} and Port_Mode_{2} are reserved and are not used.

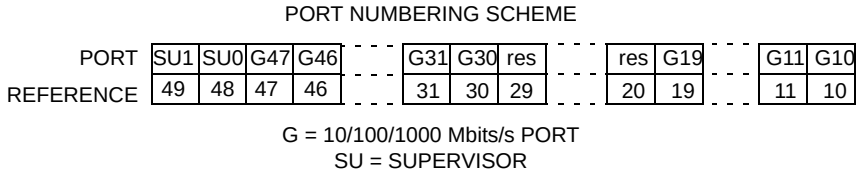


Figure 200. Port Numbering Scheme

Appendix A: Registers (continued)

Priority_Decode_Table_{0..4}

Description: Decodes the 4-bit encoded priority value prior to transmission.

Table 265. Priority_Decode_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_df00
Register Size	128
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	32
Record Spacing	4

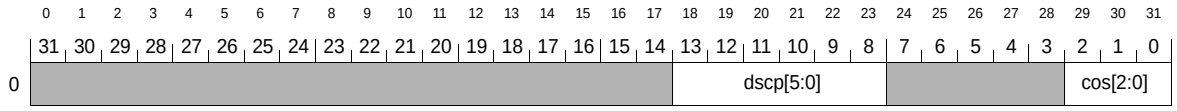


Figure 201. Priority_Decode_Table_{0..4} Register Diagram

Table 266. Priority_Decode_Table_{0..4} Field Parameters

Field Name	Parameter	Description
dscp_{}{0..31}[5:0]	Mode = R/W Offset = 0.18 Instances = 1	The decoded DSCP value.
cos_{}{0..31}[2:0]	Mode = R/W Offset = 0.29 Instances = 1	The decoded CoS value.

During packet reception, a 4-bit priority value is assigned to each packet. During packet transmission, this table is used to convert that 4-bit priority value back to either a 3-bit CoS field (for use in the VLAN tag) or a 6-bit DSCP value (for use in the IP header). This table is addressed by the concatenation of the packet's demote status and its priority value computed during ingress processing.

Note: Priority_Decode_Table_{0} and Priority_Decode_Table_{2} are reserved and are not used.

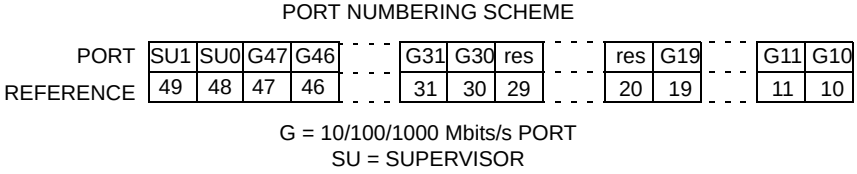


Figure 202. Port Numbering Scheme

Appendix A: Registers (continued)

Priority_Encode_Table_{0..4}

Description: Encodes the DSCP field to an internal priority representation.

Table 267. Priority_Encode_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_de00
Register Size	256
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	64
Record Spacing	4

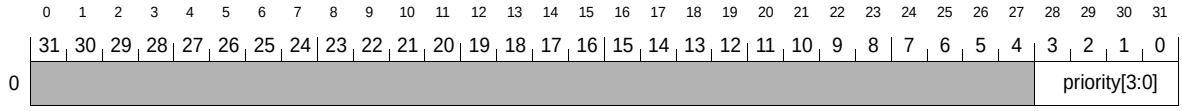


Figure 203. Priority_Encode_Table_{0..4} Register Diagram

Table 268. Priority_Encode_Table_{0..4} Field Parameters

Field Name	Parameter	Description
priority[3:0]	Mode = R/W Offset = 0.28 Instances = 1	The encoded priority value.

This table is used to encode the IP header's 6-bit DSCP value. This table is addressed by the DSCP value extracted from the packet's IP header, and the 4-bit encoded priority value is returned.

Note: Priority_Encode_Table_{0} and Priority_Encode_Table_{2} are reserved and are not used.

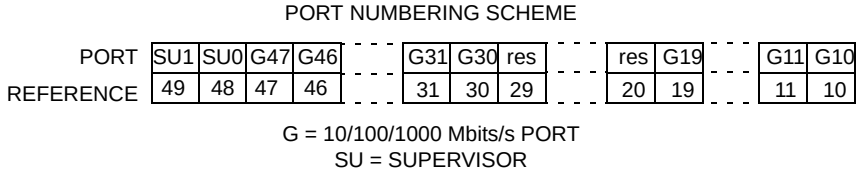


Figure 204. Port Numbering Scheme

Appendix A: Registers (continued)

Rx_Bytes

Description: Statistics counter.

Table 269. Rx_Bytes Register Parameters

Parameter	Value
Base Address	0x0004_6400
Register Size	200
Register Instances	1
Register Spacing	NA
Record Size	200
Record Instances	1
Record Spacing	NA

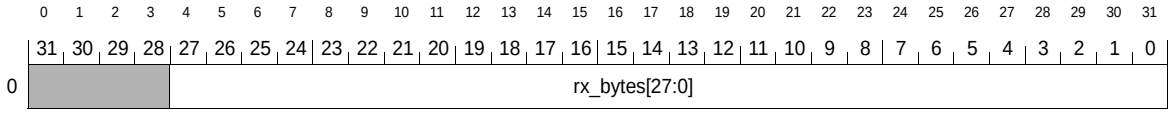


Figure 205. Rx_Bytes Register Diagram

Table 270. Rx_Bytes Field Parameters

Field Name	Parameter	Description
rx_bytes[27:0]	Mode = R/W Offset = 0.4 Instances = 28 Spacing = 4.0	The number of packet bytes received (including those with CRC errors).

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor ten times per second without risk of missing a rollover event.

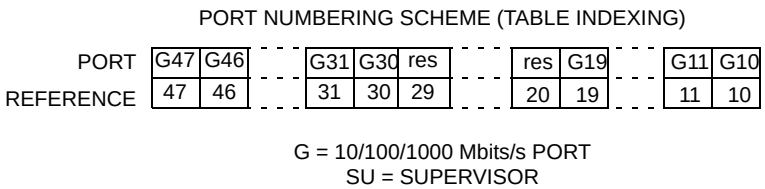


Figure 206. Port Numbering Scheme

Appendix A: Registers (continued)

Rx_Error_Packets

Description: Statistics counters.

Table 271. Rx_Error_Packets Register Parameters

Parameter	Value
Base Address	0x0004_5800
Register Size	_2
Register Instances	1
Register Spacing	NA
Record Size	800
Record Instances	1
Record Spacing	NA

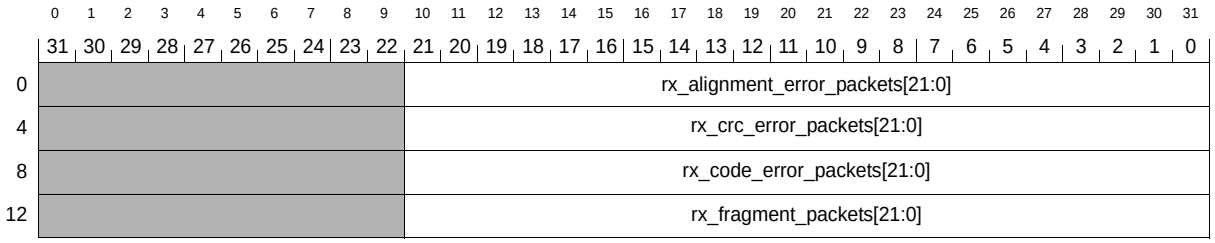


Figure 207. Rx_Error_Packets Register Diagram

Table 272. Rx_Error_Packets Field Parameters

Field Name	Parameter	Description
rx_alignment_error_packets[21:0]	Mode = R/W Offset = 0.10 Instances = 28 Spacing = 16.0	The number of packets received with an alignment error (bad CRC and dribble bits) that are at least 64 bytes in length.
rx_crc_error_packets[21:0]	Mode = R/W Offset = 4.10 Instances = 28 Spacing = 16.0	The number of packets received with an CRC error, no dribble bits, and are at least 64 bytes in length.
rx_code_error_packets[21:0]	Mode = R/W Offset = 8.10 Instances = 28 Spacing = 16.0	The number of packets received with a CRC error and a code error. These packets do not have dribble bits and are at least 64 bytes in length.
rx_fragment_packets[21:0]	Mode = R/W Offset = 12.10 Instances = 28 Spacing = 16.0	The number of packets received with a CRC error and are less than 64 bytes in length. Code errors and dribble bits may also be present.

Appendix A: Registers (continued)

Rx_Error_Packets (continued)

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor ten times per second without risk of missing a rollover event.

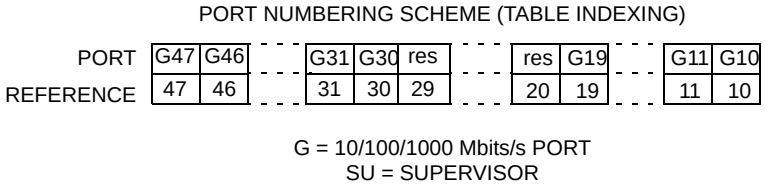


Figure 208. Port Numbering Scheme

Appendix A: Registers (continued)

Rx_Length_Histogram

Description: Statistics counters.

Table 273. Rx_Length_Histogram Register Parameters

Parameter	Value
Base Address	0x0004_4000
Register Size	2000
Register Instances	1
Register Spacing	NA
Record Size	2000
Record Instances	1
Record Spacing	NA

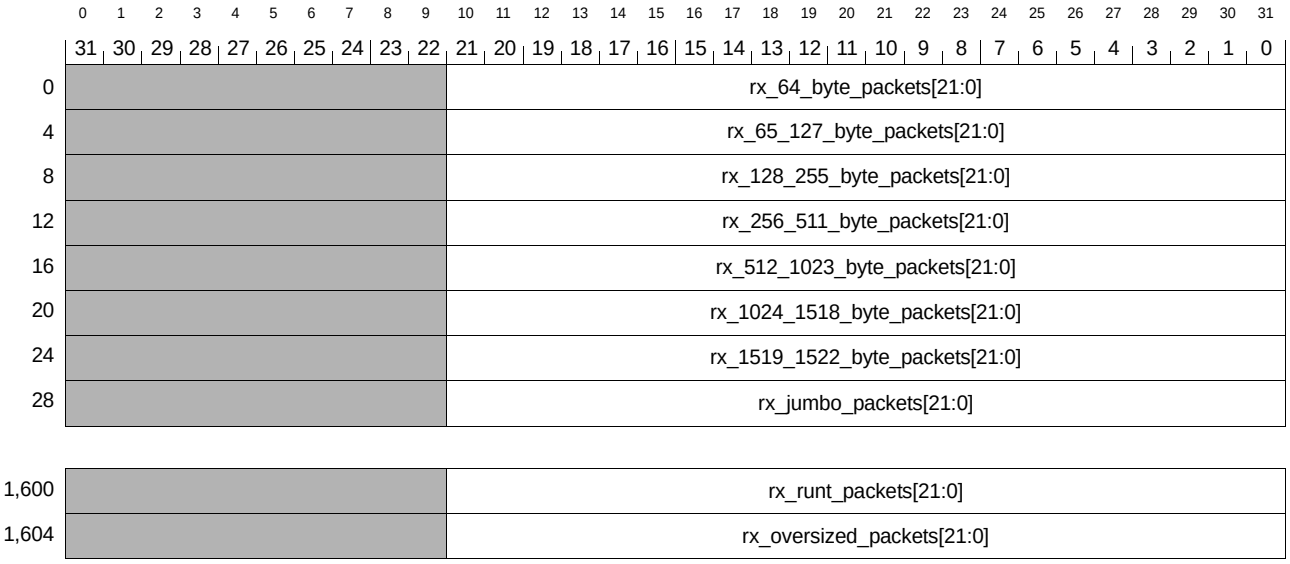


Figure 209. Rx_Length_Histogram Register Diagram

Table 274. Rx_Length_Histogram Field Parameters

Field Name	Parameter	Description
rx_64_byte_packets[21:0]	Mode = R/W Offset = 0.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is 64 bytes.
rx_65_127_byte_packets[21:0]	Mode = R/W Offset = 4.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is from 65 to 127 bytes.
rx_128_255_byte_packets[21:0]	Mode = R/W Offset = 8.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is from 128 to 255 bytes.

Appendix A: Registers (continued)

Rx_Length_Histogram (continued)

Table 274. Rx_Length_Histogram Field Parameters (continued)

Field Name	Parameter	Description
rx_256_511_byte_packets[21:0]	Mode = R/W Offset = 12.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is from 256 to 511 bytes.
rx_512_1023_byte_packets[21:0]	Mode = R/W Offset = 16.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is from 512 to 1,023 bytes.
rx_1024_1518_byte_packets[21:0]	Mode = R/W Offset = 20.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is from 1,024 to 1,518 bytes.
rx_1519_1522_byte_packets[21:0]	Mode = R/W Offset = 24.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is from 1,519 to 1,522 bytes.
rx_jumbo_packets[21:0]	Mode = R/W Offset = 28.10 Instances = 48 Spacing = 32	Packets of all types received without errors whose overall length is greater than 1,522 bytes and less than 16,384 bytes.
rx_runt_packets[21:0]	Mode = R/W Offset = 1600.10 Instances = 48 Spacing = 8	Packets of all types received without errors whose overall length is less than 64 bytes.
rx_oversized_packets[21:0]	Mode = R/W Offset = 1604.10 Instances = 48 Spacing = 8	Packets of all types whose overall length is greater than or equal to 16,384 bytes.

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor ten times per second without risk of missing a rollover event.

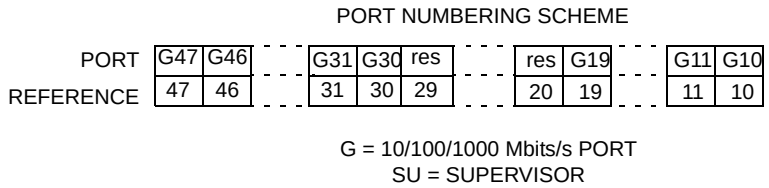


Figure 210. Port Numbering Scheme

Appendix A: Registers (continued)

Rx_Packets

Description: Statistics counters.

Table 275. Rx_Packets Register Parameters

Parameter	Value
Base Address	0x0004_5c00
Register Size	1000
Register Instances	1
Register Spacing	NA
Record Size	1000
Record Instances	1
Record Spacing	NA

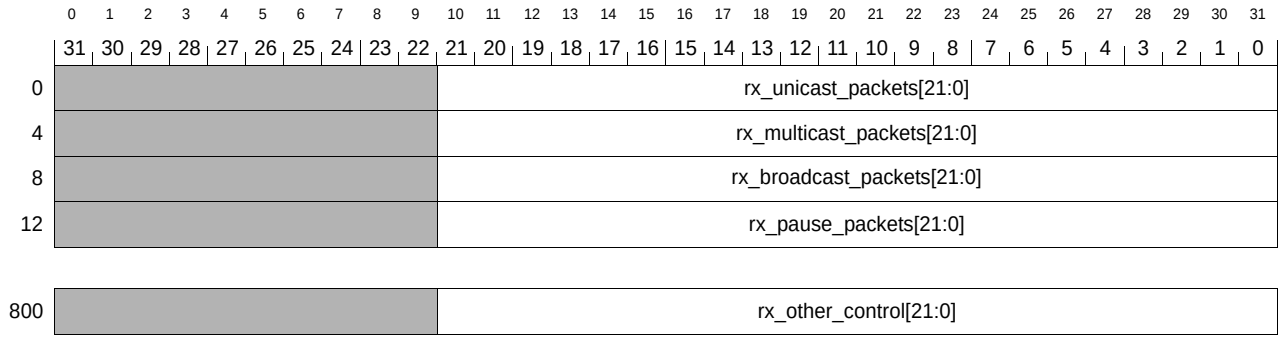


Figure 211. Rx_Packets Register Diagram

Appendix A: Registers (continued)

Rx_Packets (continued)

Table 276. Rx_Packets Field Parameters

Field Name	Parameter	Description
rx_unicast_packets[21:0]	Mode = R/W Offset = 0.10 Instances = 48 Spacing = 16.0	The number of packets received (without errors) whose destination address is a unicast address.
rx_multicast_packets[21:0]	Mode = R/W Offset = 4.10 Instances = 48 Spacing = 16.0	The number of packets received (without errors) whose destination address is a multicast address.
rx_broadcast_packets[21:0]	Mode = R/W Offset = 8.10 Instances = 48 Spacing = 16.0	The number of packets received (without errors) whose destination address is the broadcast address.
rx_pause_packets[21:0]	Mode = R/W Offset = 12.10 Instances = 48 Spacing = 16.0	The number of packets received (without errors) that are pause control packets.
rx_other_control[21:0]	Mode = R/W Offset = 800.10 Instances = 48 Spacing = 4.0	The number of packets received (without errors) that are control packets other than pause packets.

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor ten times per second without risk of missing a rollover event.

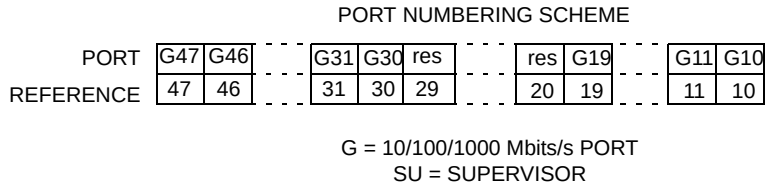


Figure 212. Port Numbering Scheme

Appendix A: Registers (continued)

Serdes_Control_{4}

Description: Provides access to SerDes registers.

Table 277. Serdes_Control_{4} Register Parameters

Parameter	Value
Base Address	0x000c_8240
Register Size	12
Register Instances	1
Register Spacing	128
Record Size	12
Record Instances	1
Record Spacing	NA

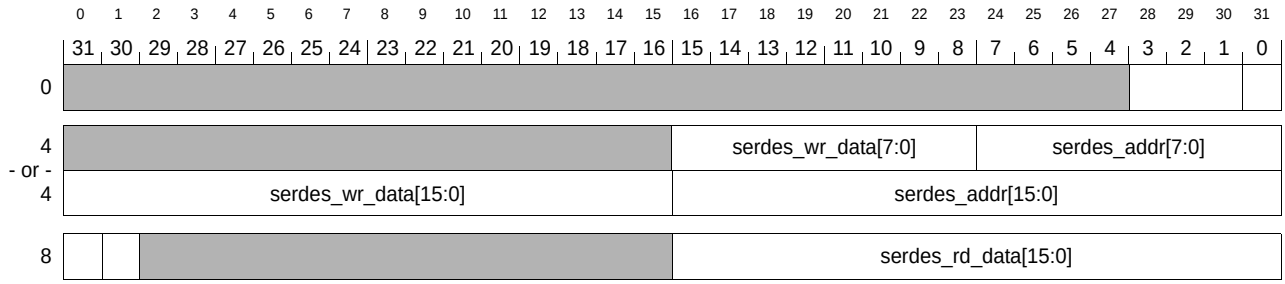


Figure 213. Serdes_Control_{4} Register Diagram

Appendix A: Registers (continued)

Serdes_Control_{4} (continued)

Table 278. Serdes_Control_{4} Field Parameters

Field Name	Parameter	Description
access_command[2:0]	Mode = WO Offset = 0.28 Instances = 1	Defines the type of SerDes register access function to be performed: 000_2 = 8-bit read (SerDes macro registers). 100_2 = 8-bit write (SerDes macro registers). 010_2 = 16-bit read (SerDes PRBS registers). 110_2 = 16-bit write (SerDes PRBS registers). $xx1_2$ = reserved.
command_start	Mode = WO Offset = 0.31 Instances = 1	Writing a one to this bit causes the execution of the specified access command.
serdes_wr_data[7:0] or serdes_wr_data[15:0]	Mode = R/W Offset = 4.16 Instances = 1 or Mode = R/W Offset = 4.0 Instances = 1	The 8-bit or 16-bit data word to be written to the specified SerDes register. The width and offset of this field must be consistent with the command issued via `access_command[2:0]`.
serdes_wr_addr[7:0] or serdes_wr_addr[15:0]	Mode = R/W Offset = 4.24 Instances = 1 or Mode = R/W Offset = 4.16 Instances = 1	The 8-bit or 16-bit address to be use to specify the desired SerDes register. The width and offset of this field must be consistent with the command issued via `access_command[2:0]`.
serdes_command_busy	Mode = RO Offset = 8.0 Instances = 1	This bit is asserted to indicate that the requested SerDes registers access command is currently being executed. While this bit is asserted, writes to all fields of this register are ignored.
serdes_command_done	Mode = RO Offset = 8.1 Instances = 1	This bit is asserted to indicate that the requested SerDes register access command has completed operation. If the requested command was a read command, then this bit serves as an indication that `serdes_rd_data[15:0]` is valid.
serdes_rd_data[15:0]	Mode = RO Offset = 8.16 Instances = 1	Data from read commands is returned via this field. For 8-bit reads, the data occupies the least significant 8 bits of this field `(serdes_rd_data[7:0])`.

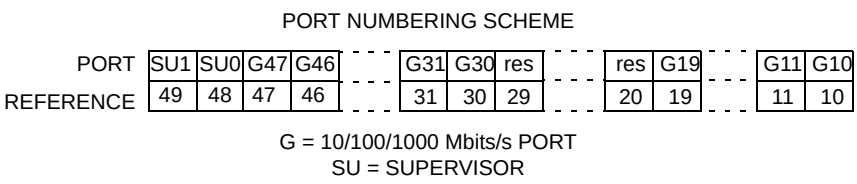


Figure 214. Port Numbering Scheme

Appendix A: Registers (continued)

Supervisor_Endian

Description: Establishes the endian mode of the *PCI* interface.

Table 279. Supervisor_Endian Register Parameters

Parameter	Value
Base Address	0x000c_c520
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

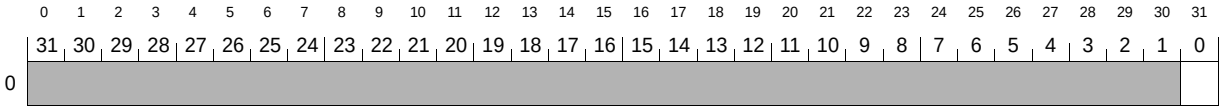


Figure 215. Supervisor_Endian Register Diagram

Table 280. Supervisor_Endian Field Parameters

Field Name	Parameter	Description
big_endian	Mode = R/W Offset = 0.31 Instances = 1	When asserted, this *PCI* interface presents data across the *PCI* bus in a big-endian fashion. Little endian (this bit deasserted) is the default behavior.

Appendix A: Registers (continued)

Supervisor_Ind

Description: Provides supervisor indications from `supervisor_access`.

Table 281. Supervisor_Ind Register Parameters

Parameter	Value
Base Address	0x000c_c448
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

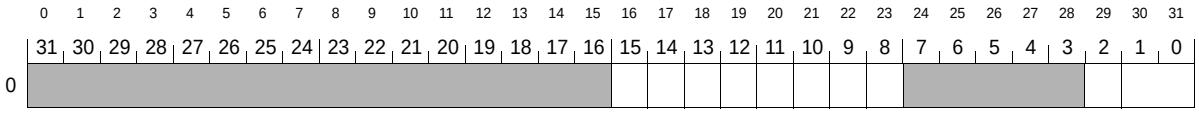


Figure 216. Supervisor_Ind Register Diagram

Table 282. Supervisor_Ind Field Parameters

Field Name	Parameter	Description
rx_fifo_not_empty_{7..0}	Mode = RO Offset = 0.16 Instances = 8 Spacing = 0.1	These bits indicate when the various supervisor receive FIFOs are not empty. Each bit corresponds to a single FIFO. In order to clear one of these bits, its associated FIFO must be emptied by the supervisor.
invalid_addr	Mode = R/W Offset = 0.29 Instances = 1	This bit is asserted to indicate that a recent supervisor access to the ET3028-50 was not addressed to a valid location. The address causing this indication can be determined via `Supervisor_Invalid_Addr`. `invalid_addr` is cleared by the supervisor writing a one to this bit location.
tx_packet[1:0]	Mode = R/W Offset = 0.30 Instances = 1	This bit is asserted to indicate that a marked packet (one whose `Supervisor_Tx_Descriptor.indicate` bit is asserted) has been transferred to the ET3028-50 for transmission. Bits in `tx_packet[1:0]` are cleared by the supervisor writing a one to their bit locations.

Various indications to the supervisor regarding supervisor. These indications may be masked by `Supervisor_Ind_Mask`.

Appendix A: Registers (continued)

Supervisor_Ind_Mask

Description: `supervisor_access`-related indication mask.

Table 283. Supervisor_Ind_Mask Register Parameters

Parameter	Value
Base Address	0x000c_c44c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

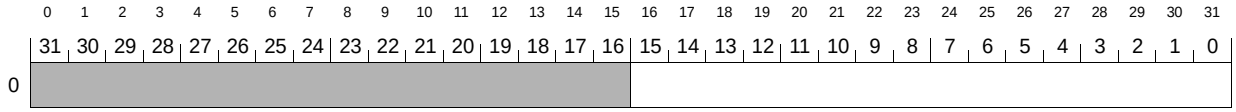


Figure 217. Supervisor_Ind_Mask Register Diagram

Table 284. Supervisor_Ind_Mask Field Parameters

Field Name	Parameter	Description
ind_mask[15:0]	Mode = RO Offset = 0.16 Instances = 1	Various indication mask bits.

This register shows the current state of the indication mask. Mask bits are set via `Supervisor_Ind_Mask_Set` and cleared via `Supervisor_Ind_Mask_Clear`.

For the definitions of the various mask bits, see `Supervisor_Ind`, page 203.

Appendix A: Registers (continued)

Supervisor_Ind_Mask_Clear

Description: Clears bits in Supervisor_Ind_Mask.

Table 285. Supervisor_Ind_Mask_Clear Register Parameters

Parameter	Value
Base Address	0x000c_c450
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

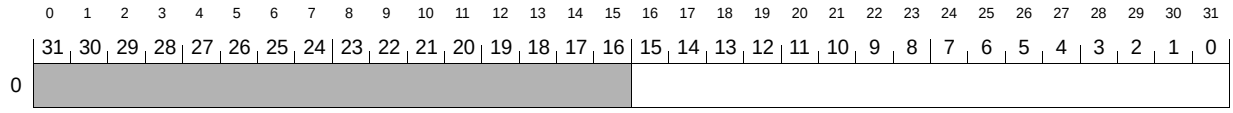


Figure 218. Supervisor_Ind_Mask_Clear Register Diagram

Table 286. Supervisor_Ind_Mask_Clear Field Parameters

Field Name	Parameter	Description
ind_mask_clear[15:0]	Mode = WO Offset = 0.16 Instances = 1	Various indication mask clear bits.

Writing ones to bit locations in this register causes the corresponding bits in Supervisor_Ind_Mask to be cleared.

For the definitions of the various mask clear bits, see Supervisor_Ind, page 203.

Appendix A: Registers (continued)

Supervisor_Ind_Mask_Set

Description: Sets bits in `Supervisor_Ind_Mask`.

Table 287. Supervisor_Ind_Mask_Set Register Parameters

Parameter	Value
Base Address	0x000c_c454
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

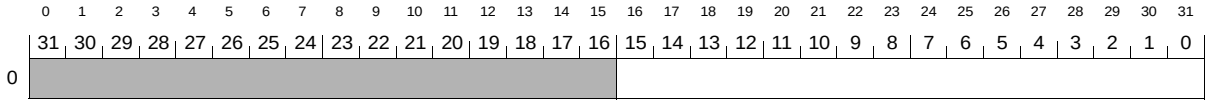


Figure 219. Supervisor_Ind_Mask_Set Register Diagram

Table 288. Supervisor_Ind_Mask_Set Field Parameters

Field Name	Parameter	Description
ind_mask_set[15:0]	Mode = WO Offset = 0.16 Instances = 1	Various indication mask set bits.

Writing ones to bit locations in this register causes the corresponding bits in `Supervisor_Ind_Mask` to be set.

For the definitions of the various mask set bits, see `Supervisor_Ind`, page 203.

Appendix A: Registers (continued)

Supervisor_Int

Description: Supervisor-related interrupts.

Table 289. Supervisor_Int Register Parameters

Parameter	Value
Base Address	0x000c_c458
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

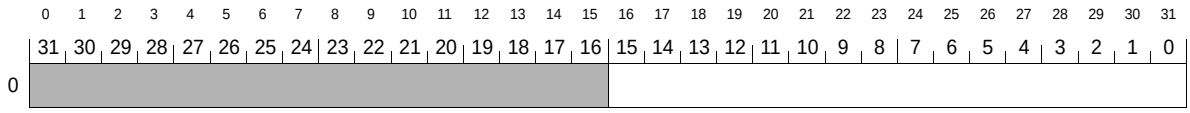


Figure 220. Supervisor_Int Register Diagram

Table 290. Supervisor_Int Field Parameters

Field Name	Parameter	Description
int[15:0]	Mode = RO Offset = 0.16 Instances = 1	Various supervisor interrupts.

This register presents those indications that have not been masked by `Supervisor_Int_Mask`.

For the definitions of the various interrupt bits, see `Supervisor_Ind`, page 203.

Appendix A: Registers (continued)

Supervisor_Int_Mask

Description: `supervisor_access`-related interrupt mask.

Table 291. Supervisor_Int_Mask Register Parameters

Parameter	Value
Base Address	0x000c_c45c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

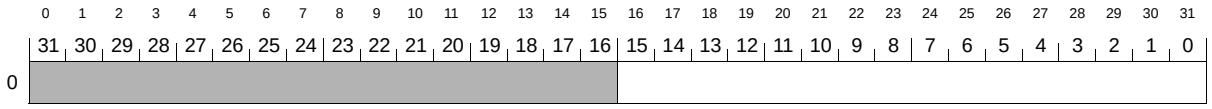


Figure 221. Supervisor_Int_Mask Register Diagram

Table 292. Supervisor_Int_Mask Field Parameters

Field Name	Parameter	Description
int_mask[15:0]	Mode = RO Offset = 0.16 Instances = 1	Various interrupt mask bits.

This register shows the current state of the indication mask. Mask bits are set via `Supervisor_Int_Mask_Set` and cleared via `Supervisor_Int_Mask_Clear`.

For the definitions of the various mask bits, see `Supervisor_Ind`, page 203.

Appendix A: Registers (continued)

Supervisor_Int_Mask_Clear

Description: Clears bits in Supervisor_Int_Mask.

Table 293. Supervisor_Int_Mask_Clear Register Parameters

Parameter	Value
Base Address	0x000c_c460
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

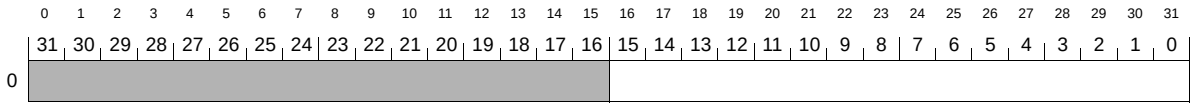


Figure 222. Supervisor_Int_Mask_Clear Register Diagram

Table 294. Supervisor_Int_Mask_Clear Field Parameters

Field Name	Parameter	Description
int_mask_clear[15:0]	Mode = WO Offset = 0.16 Instances = 1	Various interrupt mask clear bits.

Writing ones to bit locations in this register causes the corresponding bits in Supervisor_Int_Mask to be cleared.

For the definitions of the various mask clear bits, see Supervisor_Ind, page 203.

Appendix A: Registers (continued)

Supervisor_Int_Mask_Set

Description: Sets bits in `Supervisor_Int_Mask`.

Table 295. Supervisor_Int_Mask_Set Register Parameters

Parameter	Value
Base Address	0x000c_c464
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

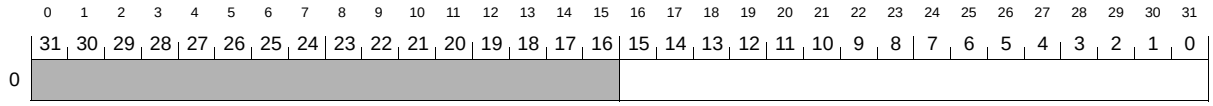


Figure 223. Supervisor_Int_Mask_Set Register Diagram

Table 296. Supervisor_Int_Mask_Set Field Parameters

Field Name	Parameter	Description
int_mask_set[15:0]	Mode = WO Offset = 0.16 Instances = 1	Various interrupt mask set bits.

Writing ones to bit locations in this register causes the corresponding bits in `Supervisor_Int_Mask` to be set.

For the definitions of the various mask set bits, see `Supervisor_Ind`, page 203.

Appendix A: Registers (continued)

Supervisor_Invalid_Addr

Description: Provides supervisor with the address that resulted in an invalid address indication.

Table 297. Supervisor_Invalid_Addr Register Parameters

Parameter	Value
Base Address	0x000c_c468
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

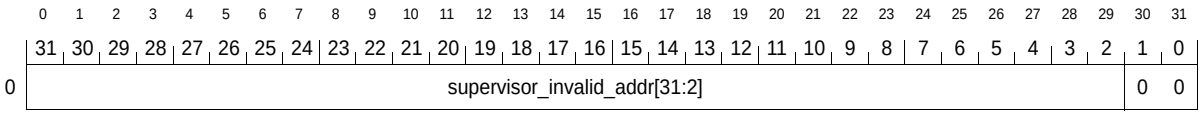


Figure 224. Supervisor_Invalid_Addr Register Diagram

Table 298. Supervisor_Invalid_Addr Field Parameters

Field Name	Parameter	Description
supervisor_invalid_addr[31:2]	Mode = R/W Offset = 0.0 Instances = 1	The location of an invalid address access by the supervisor.

This register captures and presents the address of a supervisor access cycle that failed to address a valid location. Once an address value is captured, this register ignores all future invalid address events until `Supervisor_Ind.invalid_addr` has been cleared.

Appendix A: Registers (continued)

Supervisor_Maximum_Packet_Length

Description: Determines the maximum packet length expected to be received by the supervisor.

Table 299. Supervisor_Maximum_Packet_Length Register Parameters

Parameter	Value
Base Address	0x000c_c540
Register Size	32
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	8
Record Spacing	4

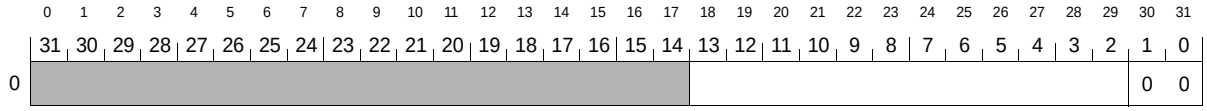


Figure 225. Supervisor_Maximum_Packet_Length Register Diagram

Table 300. Supervisor_Maximum_Packet_Length Field Parameters

Field Name	Parameter	Description
maximum_packet_length_{0..7}[13:2]	Mode = R/W Offset = 0.18 Instances = 1	Maximum packet length. **Note:** For proper operation, the minimum value for this field should be 0x4 or greater.

The `maximum_packet_length_{n}[13:2]` value is used to determine where to truncate packets and when it is necessary to wrap from the end (or near the end) of a receive FIFO to the start.

If a received packet's length exceeds `maximum_packet_length[13:2]`, then the packet is truncated to a length equal to `maximum_packet_length[13:2]` during its transfer from the ET3028-50 to the supervisor's memory system.

Wrapping from the end of a receive FIFO space to the start occurs when the space remaining in a receive FIFO is less than the value of `maximum_packet_length[13:2]`.

There are eight records in this register. Each record corresponds to an individual supervisor receive queue according to the following table.

Appendix A: Registers (continued)

Supervisor_Maximum_Packet_Length (continued)

Table 301. Supervisor Receive Queue and Corresponding Register Records

Queue	Register Record
400	0
401	1
402	2
403	3
404	4
405	5
406	6
407	7

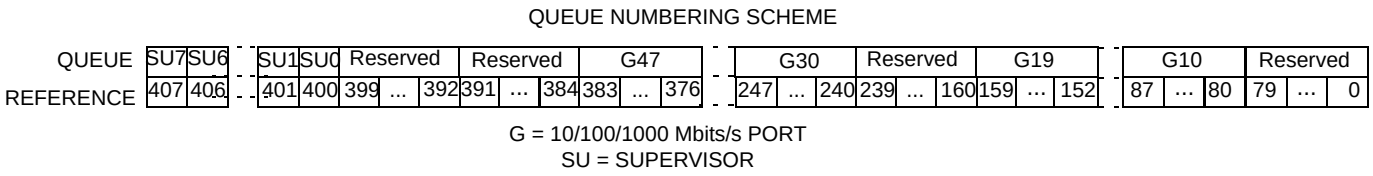


Figure 226. Queue Numbering Scheme

Appendix A: Registers (continued)

Supervisor_Rx_Fifo_Limits

Description: Defines the dimensions of the supervisor's receive packet FIFOs.

Table 302. Supervisor_Rx_Fifo_Limits Register Parameters

Parameter	Value
Base Address	0x000c_c480
Register Size	128
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	8
Record Spacing	16

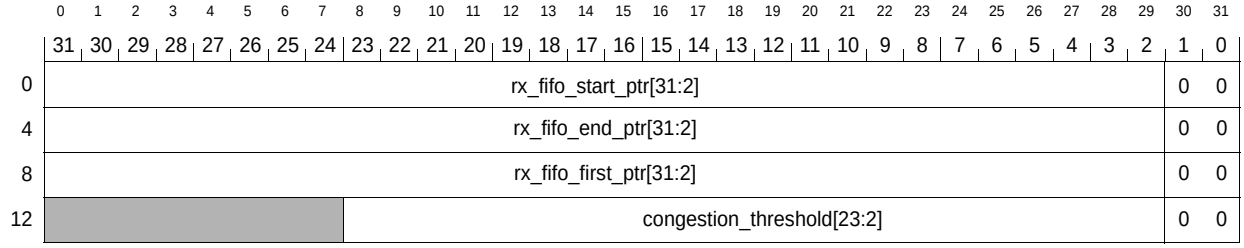


Figure 227. Supervisor_Rx_Fifo_Limits Register Diagram

Table 303. Supervisor_Rx_Fifo_Limits Field Parameters

Field Name	Parameter	Description
rx_fifo_start_ptr_{0..7}[31:2]	Mode = R/W Offset = 0.0 Instances = 1	Defines the location within supervisor memory of the first 32-bit word of a receive FIFO.
rx_fifo_end_ptr_{0..7}[31:2]	Mode = R/W Offset = 4.0 Instances = 1	Defines the location within supervisor memory of the last 32-bit word of a receive FIFO.
rx_fifo_first_ptr_{0..7}[31:2]	Mode = R/W Offset = 8.0 Instances = 1	Defines the location within supervisor memory of the packet that the supervisor is currently using. This value is maintained by the supervisor and interpreted by the ET3028-50. It prevents the ET3028-50 from overwriting packets that are either currently in use or have not yet been examined by the supervisor.
congestion_threshold_{0..7}[23:2]	Mode = R/W Offset = 12.8 Instances = 1	This field is reserved. Program to 0x3FFFFF.

Appendix A: Registers (continued)

Supervisor_Rx_Fifo_Limits (continued)

This register is used to define the physical characteristics of the eight supervisor receive FIFOs. These FIFOs reside within the supervisor's memory space.

There are eight instances of the three-field record diagrammed in the previous figure. The record at offset zero corresponds to receive FIFO zero.

`rx_fifo_start_ptr[31:2]` identifies the first physical location of the receive FIFO.
`rx_fifo_end_ptr[31:2]` identifies the last physical location of the receive FIFO.
`rx_fifo_start_ptr[31:2]` must be less than `rx_fifo_end_ptr[31:2]`. These values are established during initialization and must be left static during normal operation.

As the supervisor processes each packet, it writes the supervisor memory address of the first 32-bit word of the received packet's data structure to `rx_fifo_first_ptr[31:2]`. The first 32-bit word of each packet is `Supervisor_Rx_Packet.packet_start_ptr[31:2]`. `rx_fifo_first_ptr[31:2]` has the effect of protecting this packet from being overwritten by the ET3028-50 (the ET3028-50 checks to make sure that there is sufficient room to fit a maximum length packet into a receive FIFO without overwriting the location pointed to by `rx_fifo_first_ptr[31:2])`.

There are eight records in this register. Each record corresponds to an individual supervisor receive queue according to the following table.

Table 304. Supervisor Receive Queue and Corresponding Register Records

Queue	Register Record
400	0
401	1
402	2
403	3
404	4
405	5
406	6
407	7

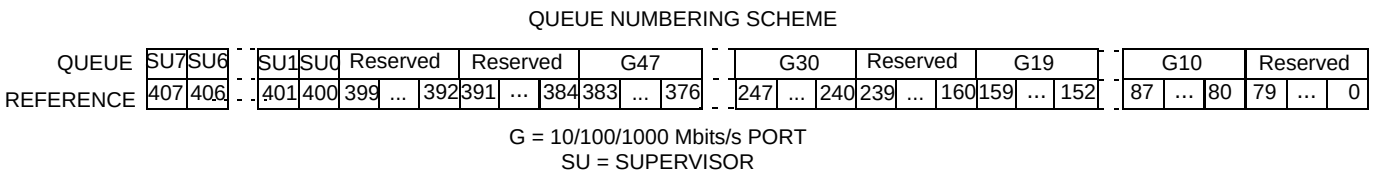


Figure 228. Queue Numbering Scheme

Appendix A: Registers (continued)

Supervisor_Rx_Fifo_Ptr

Description: The pointers used in managing the supervisor's receive packet FIFOs.

Table 305. Supervisor_Rx_Fifo_Ptr Register Parameters

Parameter	Value
Base Address	0x000c_c400
Register Size	64
Register Instances	1
Register Spacing	NA
Record Size	8
Record Instances	8
Record Spacing	8

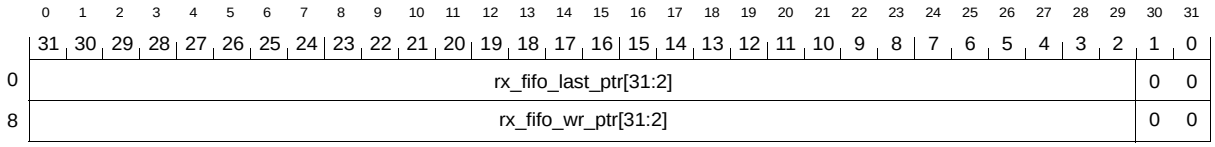


Figure 229. Supervisor_Rx_Fifo_Ptr Register Diagram

Table 306. Supervisor_Rx_Fifo_Ptr Field Parameters

Field Name	Parameter	Description
rx_fifo_last_ptr[31:2]	Mode = R/W Offset = 4.0 Instances = 1	Defines the location within supervisor memory of the last complete packet stored by the ET3028-50 in supervisor memory. This pointer is maintained by the ET3028-50 and interpreted by the supervisor. It lets the supervisor know which packet is the last in the FIFO.
rx_fifo_wr_ptr[31:2]	Mode = R/W Offset = 8.0 Instances = 1	This register is for informational purposes only and need not be accessed during normal operation. `rx_fifo_wr_ptr[31:2]` reflects the supervisor address currently or most recently addressed by the ET3028-50 during receive packet transfer operations.

Appendix A: Registers (continued)

Supervisor_Rx_Fifo_Ptr (continued)

These pointers are used during the operation of the receive FIFOs. There are eight instances of the three-field record diagrammed in the previous figure. The record at offset zero corresponds to receive FIFO zero.

As the ET3028-50 deposits packets into the FIFOs maintained within supervisor memory, it advances `rx_fifo_last_ptr[31:2]`. At all times, this pointer identifies the last complete packet deposited into the supervisor's receive FIFOs. This pointer is set to point to the first 32-bit word of a receive packet data structure `(Supervisor_Rx_Packet.packet_start_ptr[31:2])`. By interpreting this pointer, the supervisor is able to know which packet is the last one in the FIFO and, hence, when it should cease processing receive packets for the associated FIFO.

Whenever `Supervisor_Rx_Fifo_Limits.rx_fifo_first_ptr[31:2]` and `rx_fifo_last_ptr[31:2]` are equal, the corresponding FIFO is empty.

There are eight records in this register. Each record corresponds to an individual supervisor receive queue according to the following table.

Table 307. Supervisor Receive Queue and Corresponding Register Records

Queue	Register Record
400	0
401	1
402	2
403	3
404	4
405	5
406	6
407	7

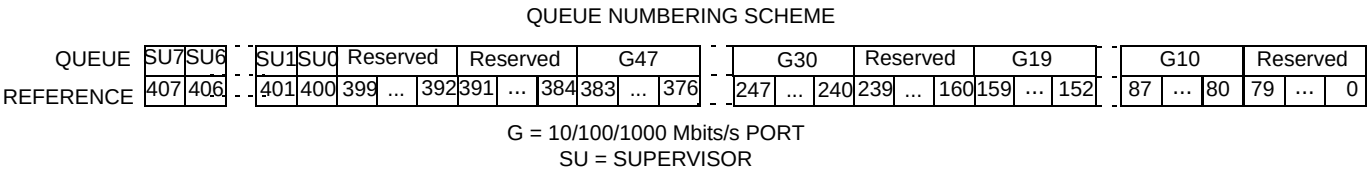


Figure 230. Queue Numbering Scheme

Appendix A: Registers (continued)

Supervisor_Statistics_Transfer_Addr

Description: Defines the starting address of the statistics transfer destination.

Table 308. Supervisor_Statistics_Transfer_Addr Register Parameters

Parameter	Value
Base Address	0x000c_c46c
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

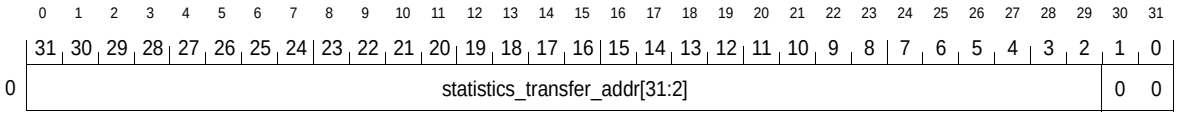


Figure 231. Supervisor_Statistics_Transfer_Addr Register Diagram

Table 309. Supervisor_Statistics_Transfer_Addr Field Parameters

Field Name	Parameter	Description
statistics_transfer_addr[31:2]	Mode = R/W Offset = 0.0 Instances = 1	Defines the location within supervisor memory to where the Ethernet MAC statistics block is to be transferred.

This register is used to define the starting location of a block of memory in the supervisor's address space that is used as the destination for Ethernet MAC statistics block transfers. The actual transfer of statistics data is initiated by writing to this register.

Appendix A: Registers (continued)

Supervisor_Statistics_Transfer_Status

Description: Indicates the status of a statistics block transfer.

Table 310. Supervisor_Statistics_Transfer_Status Register Parameters

Parameter	Value
Base Address	0x000c_c470
Register Size	4
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	1
Record Spacing	NA

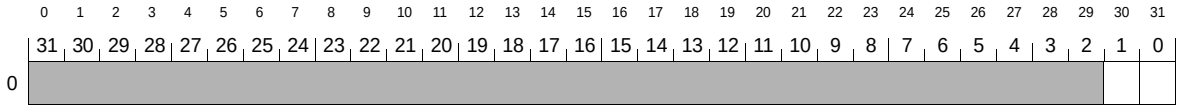


Figure 232. Supervisor_Statistics_Transfer_Status Register Diagram

Table 311. Supervisor_Statistics_Transfer_Status Field Parameters

Field Name	Parameter	Description
statistics_transfer_done	Mode = R/W Offset = 0.30 Instances = 1	This bit is asserted upon the completion of a statistics block transfer. It is reset by writing a one to its bit position.
statistics_transfer_busy	Mode = RO Offset = 0.31 Instances = 1	This bit is asserted during the transfer of the statistics block to indicate that such a transfer is in process.

This register provides basic status for statistics block transfers. A statistics block transfer is initiated by writing the block's transfer destination address to `Supervisor_Statistics_Transfer_Addr`. Upon initiation of the transfer, `statistics_transfer_busy` is automatically asserted and remains asserted throughout the duration of the transfer. At the completion of the transfer, `statistics_transfer_busy` is deasserted and `statistics_transfer_done` is asserted. `statistics_transfer_done` is also made available to the supervisor as an interrupt-generating indication. The supervisor must write a one to the bit position of `statistics_transfer_done` in order to deassert the bit and cancel its indication.

Appendix A: Registers (continued)

Supervisor_Tx_Fifo_Limits

Description: Defines the dimensions of the supervisor's transmit packet FIFOs.

Table 312. Supervisor_Tx_Fifo_Limits Register Parameters

Parameter	Value
Base Address	0x000c_c500
Register Size	32
Register Instances	1
Register Spacing	NA
Record Size	16
Record Instances	2
Record Spacing	16

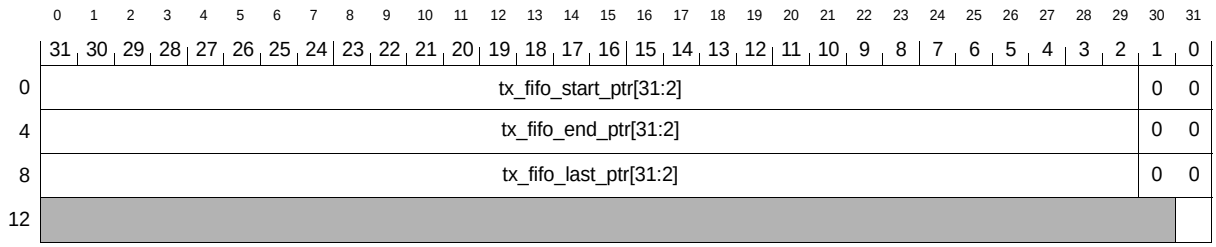


Figure 233. Supervisor_Tx_Fifo_Limits Register Diagram

Table 313. Supervisor_Tx_Fifo_Limits Field Parameters

Field Name	Parameter	Description
tx_fifo_start_ptr_{0..1}[31:2]	Mode = R/W Offset = 0.0 Instances = 1	Defines the location within supervisor memory of the first 32-bit word of a transmit FIFO.
tx_fifo_end_ptr_{0..1}[31:2]	Mode = R/W Offset = 4.0 Instances = 1	Defines the location within supervisor memory of the last 32-bit word of a transmit FIFO.
tx_fifo_last_ptr_{0..1}[31:2]	Mode = R/W Offset = 8.0 Instances = 1	Defines the location within supervisor memory of the last complete packet transmission FIFO entry. This value is maintained by the supervisor and interpreted by the ET3028-50. It prevents the ET3028-50 from reading beyond the end of the FIFO.
tx_en_{0..1}	Mode = R/W Offset = 12.31 Instances = 1	Enables transmission via the corresponding queue.

This register is used to define the physical characteristics of the two supervisor transmit FIFOs. These FIFOs reside within the supervisor's memory space.

Appendix A: Registers (continued)

Supervisor_Tx_Fifo_Limits (continued)

There are two instances of the three-field record diagrammed in the previous figure. The record at offset zero corresponds to transmit FIFO zero. Transmit FIFO zero is the low-priority FIFO. Transmit FIFO one is the high-priority FIFO. If transmit FIFO one is nonempty, it is serviced until empty prior to continuing to service transmit FIFO zero.

`tx_fifo_start_ptr[31:2]` identifies the first physical location of a transmit FIFO.
`tx_fifo_end_ptr[31:2]` identifies the last physical location of a transmit FIFO.
`tx_fifo_start_ptr[31:2]` must be less than `tx_fifo_end_ptr[31:2]`. These values are established during initialization and must be left static during normal operation.

As the user adds entries to a transmit FIFO, it advances `tx_fifo_last_ptr[31:2]` to always point to the last FIFO entry. The ET3028-50 interprets this value and uses it to determine when the FIFO is empty, nonempty, or full.

Appendix A: Registers (continued)

Supervisor_Tx_Fifo_Ptr

Description: The pointers used in managing the supervisor's transmit packet FIFOs.

Table 314. Supervisor_Tx_Fifo_Ptr Register Parameters

Parameter	Value
Base Address	0x000c_c440
Register Size	8
Register Instances	1
Register Spacing	NA
Record Size	4
Record Instances	2
Record Spacing	4

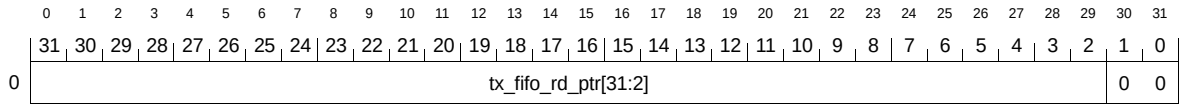


Figure 234. Supervisor_Tx_Fifo_Ptr Register Diagram

Table 315. Supervisor_Tx_Fifo_Ptr Field Parameters

Field Name	Parameter	Description
tx_fifo_rd_ptr[31:2]	Mode = R/W Offset = 4.0 Instances = 1	Defines the location within supervisor memory of the transmit FIFO entry currently being processed by the ET3028-50. This pointer is maintained by the ET3028-50 and interpreted by the supervisor.

These pointers are used during the operation of the transmit FIFOs. There are two instances of the single-field record diagrammed above. The record at offset zero corresponds to transmit FIFO zero.

As the ET3028-50 processes the packet transmit FIFO, it advances `tx_fifo_rd_ptr[31:2]`. The supervisor reads this register to determine if its next FIFO entry might overwrite the end of the FIFO. If the address of the supervisor's next FIFO write location is equal to `tx_fifo_rd_ptr[31:2]`, then that FIFO is full. When `tx_fifo_rd_ptr[31:2]` and `Supervisor_Tx_Fifo_Limits.tx_fifo_last_ptr[31:2]` are equal, then the corresponding FIFO is empty.

Appendix A: Registers (continued)

Tx_Bytes

Description: Statistics counter.

Table 316. Tx_Bytes Register Parameters

Parameter	Value
Base Address	0x0004_6500
Register Size	200
Register Instances	1
Register Spacing	NA
Record Size	200
Record Instances	1
Record Spacing	NA

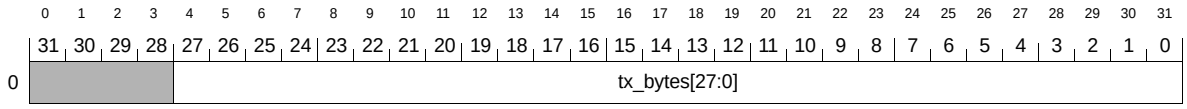


Figure 235. Tx_Bytes Register Diagram

Table 317. Tx_Bytes Field Parameters

Field Name	Parameter	Description
tx_bytes[27:0]	Mode = R/W Offset = 0.4 Instances = 48 Spacing = 4	The total number of packet bytes transmitted.

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor at ten times per second without risk of missing a rollover event.

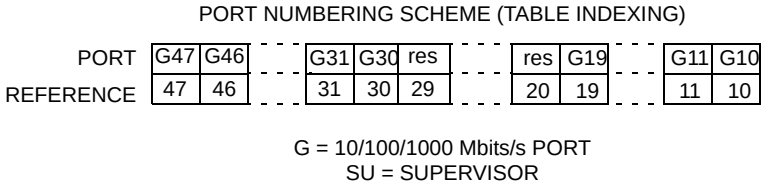


Figure 236. Port Numbering Scheme

Appendix A: Registers (continued)

Tx_Collision_Histogram

Description: Statistics counters.

Table 318. Tx_Collision_Histogram Register Parameters

Parameter	Value
Base Address	0x0004_5000
Register Size	1000
Register Instances	1
Register Spacing	NA
Record Size	1000
Record Instances	1
Record Spacing	NA

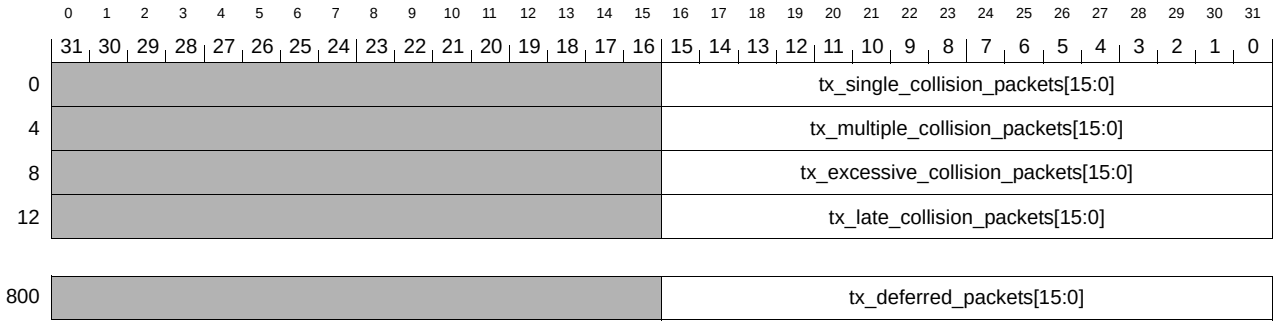


Figure 237. Tx_Collision_Histogram Register Diagram

Appendix A: Registers (continued)

Tx_Collision_Histogram (continued)

Table 319. Tx_Collision_Histogram Field Parameters

Field Name	Parameter	Description
tx_single_collision_packets[15:0]	Mode = R/W Offset = 0.16 Instances = 48 Spacing = 16.0	The number of packets successfully transmitted after experiencing one collision.
tx_multiple_collision_packets[15:0]	Mode = R/W Offset = 4.16 Instances = 48 Spacing = 16.0	The number of packets successfully transmitted after experiencing multiple collisions.
tx_excessive_collision_packets[15:0]	Mode = R/W Offset = 8.16 Instances = 48 Spacing = 16.0	The number of packets that fail to be transmitted due to excessive collisions.
tx_late_collision_packets[15:0]	Mode = R/W Offset = 12.16 Instances = 48 Spacing = 16.0	The number of packets that fail to be transmitted due to a late collision. Such packets may have experienced one or more normal collisions prior to the late collision.
tx_deferred_packets[17:0]	Mode = R/W Offset = 800.14 Instances = 48 Spacing = 4.0	A count of all packets that had to defer to either traffic on the network (half-duplex mode) or to an active pause timer (full-duplex mode).

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor at ten times per second without risk of missing a rollover event.

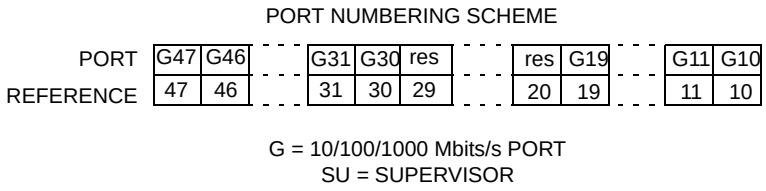


Figure 238. Port Numbering Scheme

Appendix A: Registers (continued)

Tx_Length_Histogram

Description: Statistics counters.

Table 320. Tx_Length_Histogram Register Parameters

Parameter	Value
Base Address	0x0004_4800
Register Size	1600
Register Instances	1
Register Spacing	NA
Record Size	1600
Record Instances	1
Record Spacing	NA

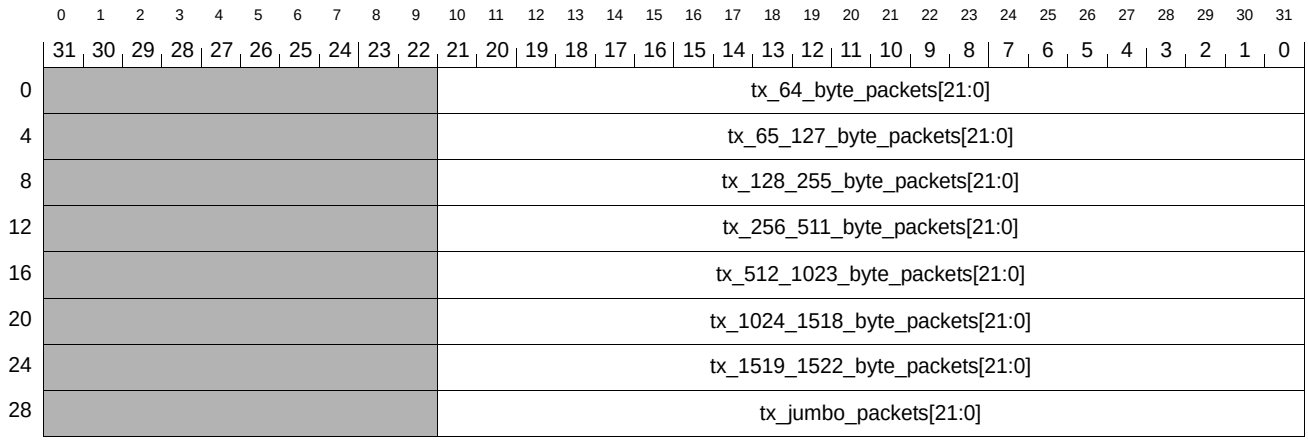


Figure 239. Tx_Length_Histogram Register Diagram

Table 321. Tx_Length_Histogram Field Parameters

Field Name	Parameter	Description
tx_64_byte_packets[21:0]	Mode = R/W Offset = 0.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is 64 bytes.
tx_65_127_byte_packets[21:0]	Mode = R/W Offset = 4.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is from 65 to 127 bytes.
tx_128_255_byte_packets[21:0]	Mode = R/W Offset = 8.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is from 128 to 255 bytes.

Appendix A: Registers (continued)

Tx_Length_Histogram (continued)

Table 321. Tx_Length_Histogram Field Parameters (continued)

Field Name	Parameter	Description
tx_256_511_byte_packets[21:0]	Mode = R/W Offset = 12.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is from 256 to 511 bytes.
tx_512_1023_byte_packets[21:0]	Mode = R/W Offset = 16.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is from 512 to 1,023 bytes.
tx_1024_1518_byte_packets[21:0]	Mode = R/W Offset = 20.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is from 1,024 to 1,518 bytes.
tx_1519_1522_byte_packets[21:0]	Mode = R/W Offset = 24.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is from 1,519 to 1,522 bytes.
tx_jumbo_packets[21:0]	Mode = R/W Offset = 28.10 Instances = 48 Spacing = 32	Packets of all types transmitted without errors whose overall length is greater than 1,522 bytes.

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor at ten times per second without risk of missing a rollover event.

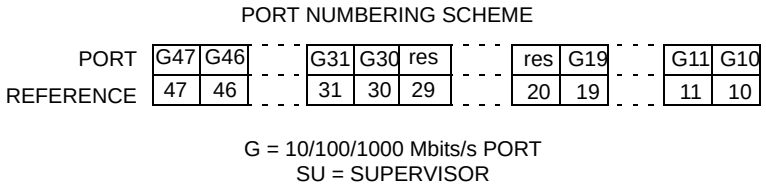


Figure 240. Port Numbering Scheme

Appendix A: Registers (continued)

Tx_Packets

Description: Statistics counters.

Table 322. Tx_Packets Register Parameters

Parameter	Value
Base Address	0x0004_6000
Register Size	800
Register Instances	1
Register Spacing	NA
Record Size	800
Record Instances	1
Record Spacing	NA

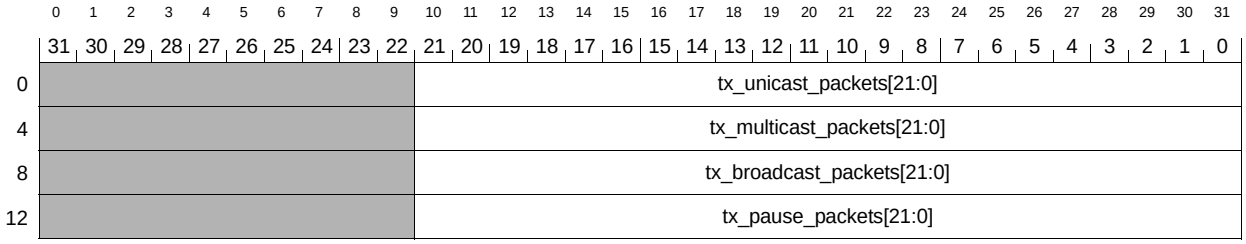


Figure 241. Tx_Packets Register Diagram

Table 323. Tx_Packets Field Parameters

Field Name	Parameter	Description
tx_unicast_packets[21:0]	Mode = R/W Offset = 0.10 Instances = 48 Spacing = 16.0	The number of unicast packets transmitted.
tx_multicast_packets[21:0]	Mode = R/W Offset = 4.10 Instances = 48 Spacing = 16.0	The number of multicast packets transmitted.
tx_broadcast_packets[21:0]	Mode = R/W Offset = 8.10 Instances = 48 Spacing = 16.0	The number of broadcast packets transmitted.
tx_pause_packets[21:0]	Mode = R/W Offset = 12.10 Instances = 48 Spacing = 16.0	The number of pause control packets transmitted.

Appendix A: Registers (continued)

Tx_Packets (continued)

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor at ten times per second without risk of missing a rollover event.

PORT NUMBERING SCHEME

PORT	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 242. Port Numbering Scheme

Appendix A: Registers (continued)

Tx_Total_Collisions

Description: Statistics counters.

Table 324. Tx_Total_Collisions Register Parameters

Parameter	Value
Base Address	0x0004_5400
Register Size	200
Register Instances	1
Register Spacing	NA
Record Size	200
Record Instances	1
Record Spacing	NA

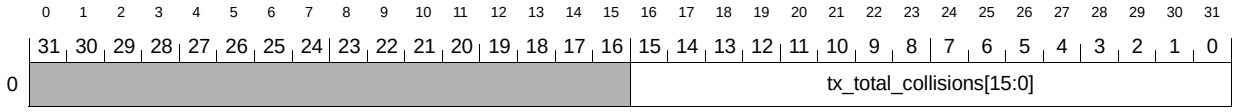


Figure 243. Tx_Total_Collisions Register Diagram

Table 325. Tx_Total_Collisions Field Parameters

Field Name	Parameter	Description
tx_total_collisions[15:0]	Mode = R/W Offset = 0.16 Instances = 48 Spacing = 4.0	A count of the total number of collisions experienced by all packets

Statistics counters roll over from their maximum count to zero without warning or indication that a rollover has occurred. It is expected that the supervisor sample these counters often and use the delta from a previous sample to the current sample to determine the overall count delta. If the current sample has a smaller value than the previous sample (indicating a rollover), then 2^n (where n is the number of bits in the statistics counter) is added to the delta to determine the actual delta. All statistics counters' widths are scaled to allow for sampling by the supervisor at ten times per second without risk of missing a rollover event.

PORT NUMBERING SCHEME (TABLE INDEXING)

PORT	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 244. Port Numbering Scheme

Appendix A: Registers (continued)

User_Protocol_{0..4}

Description: User-specified IP protocol values.

Table 326. User_Protocol_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_dfc0
Register Size	12
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	3
Record Spacing	4

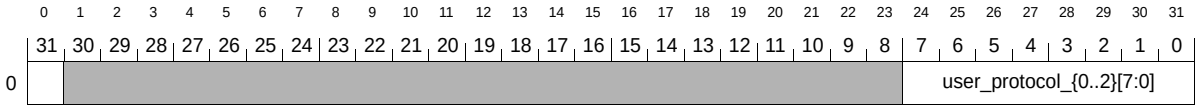


Figure 245. User_Protocol_{0..4} Register Diagram

Table 327. User_Protocol_{0..4} Field Parameters

Field Name	Parameter	Description
user_protocol_entry_valid_{}{0..2}	Mode = R/W Offset = 0.0 Instances = 1	Must be asserted for comparison to be made.
user_protocol_{}{0..2}[7:0]	Mode = R/W Offset = 0.24 Instances = 1	User-specified protocol value.

If a received packet's IP protocol value matches one of the three in this table, then its protocol is designated as one of the user defined protocols used to access the acl_protocol[2:0] value in the Acl_Protocol_Table register as part of the ACE map look-up. In Table 79 of the Acl_Protocol_Table register, the user-defined protocols are designated as user_protocol_0, user_protocol_1, and user_protocol_2, which correspond to the user_protocol_{}{0..2}[7:0] field in the above table.

Note: The User_Protocol_{0} and User_Protocol_{2} registers are reserved and are not used.

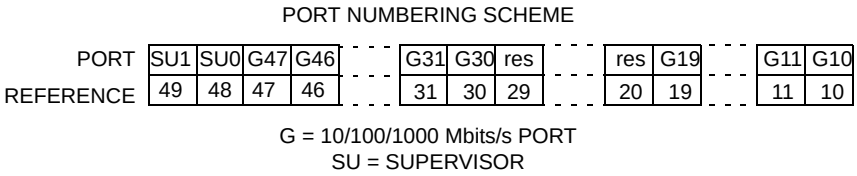


Figure 246. Port Numbering Scheme

Appendix A: Registers (continued)

User_Type_{0..4}

Description: User-specified Ethertype values.

Table 328. User_Type_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_dfd0
Register Size	12
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	3
Record Spacing	4

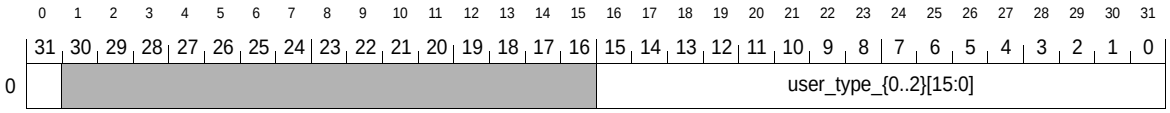


Figure 247. User_Type_{0..4} Register Diagram

Table 329. User_Type_{0..4} Field Parameters

Field Name	Parameter	Description
user_type_entry_valid_{}{0..2}	Mode = R/W Offset = 0.0 Instances = 1	Must be asserted for comparison to be made.
user_type_{}{0..2}[15:0]	Mode = R/W Offset = 0.16 Instances = 1	User-specified Ethertype value.

If a received packet's Ethertype value matches one of the three in this table, then its Ethertype is designated as one of the user-defined types used to access the acl_protocol[2:0] value in the Acl_Protocol_Table register as part of the ACE map look-up. In Table 79 of the Acl_Protocol_Table register, the user-defined types are designated as user_type_0, user_type_1, and user_type_2, which correspond to the user_type_{}{0..2}[15:0] field in the above table.

Note. The User_Type_{0} and User_Type_{2} registers are reserved and are not used.

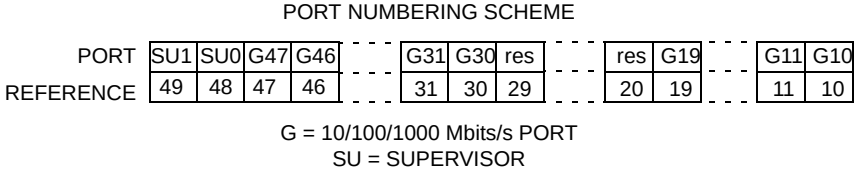


Figure 248. Port Numbering Scheme

Appendix A: Registers (continued)

Vlan_Id_Table_{0..4}

Description: This table converts VLAN indexes into VLAN identifiers.

Table 330. Vlan_Id_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_d000
Register Size	1024
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	256
Record Spacing	4

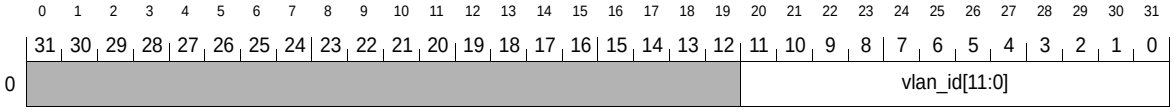


Figure 249. Vlan_Id_Table_{0..4} Register Diagram

Table 331. Vlan_Id_Table_{0..4} Field Parameters

Field Name	Parameter	Description
vlan_id[11:0]	Mode = R/W Offset = 0.20 Instances = 1	The VLAN identifier as selected by the VLAN index.

One of 256 VLAN indexes are assigned to each packet during ingress processing. This table is used to convert the 8-bit index back into a 12-bit VLAN identifier immediately prior to transmission.

Note: Vlan_Id_Table_{0} and Vlan_Id_Table_{2} are reserved and are not used.

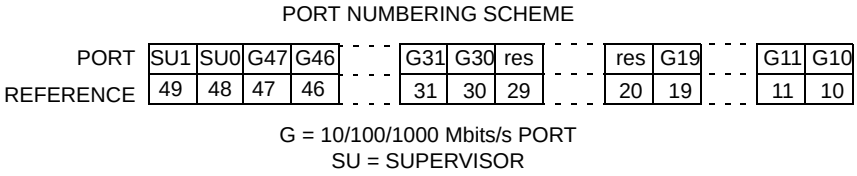


Figure 250. Port Numbering Scheme

Appendix A: Registers (continued)

Vlan_Index_Table_{0..4}

Description: This table converts VLAN identifiers into VLAN indexes.

Table 332. Vlan_Index_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_8000
Register Size	16384
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	4096
Record Spacing	4

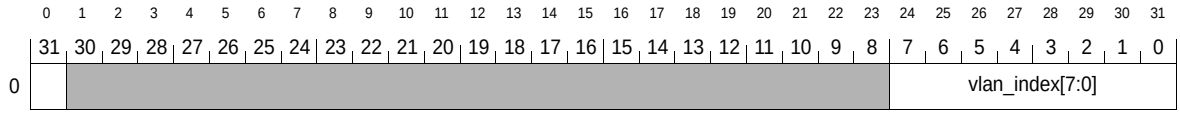


Figure 251. Vlan_Index_Table_{0..4} Register Diagram

Table 333. Vlan_Index_Table_{0..4} Field Parameters

Field Name	Parameter	Description
vlan_index_valid	Mode = R/W Offset = 0.0 Instances = 1	This bit is asserted to indicate that the submitted VLAN identifier value has a valid index stored in the table.
vlan_index[7:0]	Mode = R/W Offset = 0.20 Instances = 1	The VLAN index as selected by the VLAN identifier.

This table is used to convert the 11-bit VLAN identifier extracted from a received packet into an 8-bit VLAN index for use throughout the remainder of ingress processing. This table is addressed by the VLAN value extracted from the received packets.

The VLAN identifier space consists of 4,094 values (4,095 for revisions B1 and C). Only 256 of them may be utilized simultaneously by the ET3028-50 system. Consequently, only as many as 256 entries in this table may be marked as valid through the assertion of `vlan_index_valid`. All remaining table entries must have this bit deasserted. If `vlan_index_valid` is false, `vlan_index[7:0]` is invalid.

Care must be taken when configuring these table to ensure consistency from one packet processor to the next.

Note: Vlan_Index_Table_{0} and Vlan_Index_Table_{2} are reserved and are not used.

PORT NUMBERING SCHEME

PORT	SU1	SU0	G47	G46		G31	G30	res		res	G19		G11	G10
REFERENCE	49	48	47	46		31	30	29		20	19		11	10

G = 10/100/1000 Mbits/s PORT
SU = SUPERVISOR

Figure 252. Port Numbering Scheme

Appendix A: Registers (continued)

Vlan_Port_Protocol_Table_{0..4}

Description: This table converts port numbers and protocol indexes into VLAN indexes.

Table 334. Vlan_Port_Protocol_Table_{0..4} Register Parameters

Parameter	Value
Base Address	0x0005_dc00
Register Size	320
Register Instances	5
Register Spacing	32768
Record Size	4
Record Instances	80
Record Spacing	4

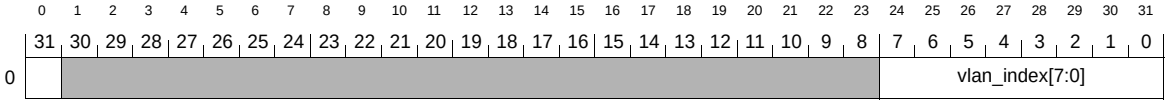


Figure 253. Vlan_Port_Protocol_Table_{0..4} Register Diagram

Table 335. Vlan_Port_Protocol_Table_{0..4} Field Parameters

Field Name	Parameter	Description
entry_valid	Mode = R/W Offset = 0.0 Instances = 1	This bit is asserted to indicate that the associated VLAN index value is valid.
vlan_index[7:0]	Mode = R/W Offset = 0.20 Instances = 1	The VLAN identifier as selected by the packet's protocol.

A 4-bit port number (range: 0..9) and a 3-bit protocol index are concatenated together to form an address in this table. The value returned is an `entry_valid` indication and a VLAN index for use in all ingress and egress packet processing. If `entry_valid` is false, `vlan_index[7:0]` is invalid.

Care must be taken when configuring these tables to ensure consistency from one packet processor to the next (e.g., avoid the accidental or inconsistent reuse of VLAN index values).

This table is addressed by a concatenation of `port[5:0]  10` and `ethertype_index[2:0]`.

Note: Vlan_Port_Protocol_Table_{0} and Vlan_Port_Protocol_Table_{2} are reserved and are not used.

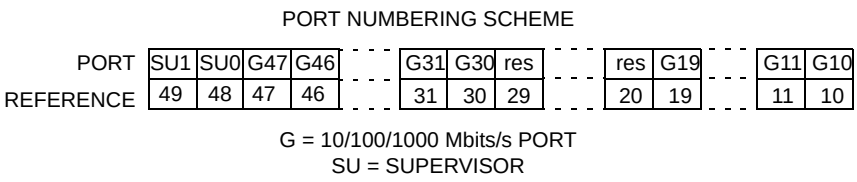


Figure 254. Port Numbering Scheme

Appendix B: Configuration

This chapter describes the methods for configuring the various features of the ET3028-50.

General

The supervisor's access to the various registers within the ET3028-50 is via a 32-bit *PCI* bus. Many of the fields and records that must be configured by the supervisor are wider than 32 bits. In general, the updating of a partial word or record may cause unpredictable behavior. To prevent this, the ET3028-50 includes line caches where necessary to ensure that integral records are always written to these registers.

Packet Buffer

The packet buffer in the ET3028-50 is self-configuring. The configuration process is initiated by the supervisor by simply asserting the `free_buffer_initialization_start` and `free_descriptor_initialization_start` bits in the `Packet_Buffer_Free_Buffer_Control` and `Packet_Buffer_Free_Descriptor_Control` registers, respectively. Upon completion of the initialization sequence, the ET3028-50 asserts the `free_buffer_initialization_done` and `free_descriptor_initialization_done` bits, accordingly.

Ethernet Interfaces

The ET3028-50 includes 28 10/100/1000 Mbits/s Ethernet ports. Various aspects of each of these interfaces must be properly configured in order to achieve normal operation.

Media Access Controllers

The media access controllers (MACs) are primarily configured via the `Mac_Mode` registers. There are five such registers: `Mac_Mode_{0..4}` with each having 10 records, one for each MAC associated with the a register.

Port Enable. Each Ethernet MAC may be completely disabled by deasserting the corresponding `port_en` bit in the `Mac_Mode_{0..4}` registers.

Receive Enable. The Ethernet MACs are disabled from receiving any network traffic unless their corresponding `gmac_rx_en_{0..9}` or `xgmac_rx_en` bit is asserted.

Speed Mode. Generally, the speed of the Ethernet ports is established automatically via autonegotiation with the system at the far end of the Ethernet link. The current speed setting is available to the supervisor via `gmac_port_speed_{0..9}[1:0]`. The valid values of this field are defined in the following table.

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

Table 336. Port Speed Indications

gmac_port_speed_{0..9}[1:0]	Port Speed
00_2	10 Mbits/s
01_2	100 Mbits/s
10_2	1 Gbit/s
11_2	Reserved

Autonegotiation. Autonegotiation is used to establish the capabilities of the systems at either end of an Ethernet link. Once the capabilities have been determined, the highest common capability is chosen for the two link partners to share.

In order to enable autonegotiation, the `auto_negotiate_en_{0..9}` bit corresponding to the appropriate Ethernet interface must be asserted.

To manually force the link to renegotiate, the `restart_auto_negotiation_{0..9}` bit is pulsed (asserted and then deasserted). The ET3028-50 detects the rising edge of this signal and immediately initiates an autonegotiation session with the corresponding link partner.

SFP Autonegotiation. For ports 44 through 47, 1000BASE-X autonegotiation is supported.

For revision C, once the appropriate `auto_negotiate_en_{0..9}` bit is asserted for ports 44 through 47, `gmac_rx_pause_en_{}{0..9}` and `gmac_flow_control_initiate_en_{}{0..9}` become the PAUSE (PS1) and ASM_DIR (PS2) bits, respectively, as defined in Table 37-2 of the *IEEE* Standard *802.3-2002.* Once autonegotiation is complete, the `gmac_rx_pause_en_{}{0..9}` and `gmac_flow_control_initiate_en_{}{0..9}` bits give the pause priority resolution for PAUSE and ASM_DIR, respectively, as defined in Table 37-4 of the standard.

Autonegotiation Override. If desired, autonegotiation may be disabled and overridden with manually prescribed parameters. By asserting `good_link_force_{0..9}`, the supervisor disables autonegotiation and enables the use of `speed_mode_force_{0..9}[1:0]` and `full_duplex_force_{0..9}` to set the port's operating characteristics.

The `full_duplex_force_{0..9}` bit is asserted to set the port to operate in full-duplex mode. When deasserted, this bit causes the port to operate in half duplex.

The `speed_mode_force_{0..9}[1:0]` field is used to set the corresponding port to a particular operating speed. The valid settings are defined in the following table.

Table 337. Port Speed Settings

speed_mode_force_{0..9}[1:0]	Port Speed
00_2	10 Mbits/s
01_2	100 Mbits/s
10_2	1 Gbit/s
11_2	Reserved

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

Loopback

Loopback enable bits (gmac_port_loopback_en_{0..9} or xgmac_port_loopback_en) may be asserted to place a port into a loopback mode. When in this mode, transmit packets are looped back into the device immediately prior to exiting the device via an SGMII interface. During loopback operations, no packets are actually transmitted and all receive traffic is ignored by the affected ports.

Note: The ET3028-50 does not attempt to align the transition into and out of the loopback mode with a gap between packets. Therefore, if a port is active during the transition into or out of the loopback mode, one or more transmit or receive packets may become corrupted.

Flow Control. There are two flow control actions that may be independently enabled: the initiation of flow control on a link, and the reaction to flow control on a link. The gmac_flow_control_initiate_en_{0..9} bits enable the respective MACs to transmit flow control packets when directed to do so by the packet buffer's congestion state. If this bit is deasserted, the associated MAC will not transmit flow control packets even if the packet buffer is indicating that it is congested.

For a MAC to react to the reception of a flow control packet, its gmac_rx_pause_en_{0..9} bit must be asserted. If this bit is deasserted, the corresponding MAC ignores all received flow control packets.

Flow control packets transmitted by the ET3028-50 must have a valid MAC source address. These source address values are supplied to the MACs via the mac_src_addr[47:6] field of the Mac_Global_Mode register. These upper bits of the MAC source address are shared by all of the MACs. The least significant 6 bits of the 48-bit MAC source address are hardwired to each MAC and are equivalent to the corresponding port number (0 through 47).

Finally, the MAC pause quanta must be specified. This value is included in the transmitted flow control packets to tell the far-end system how long to wait prior to resuming transmission. Each unit of pause is equal to 512 bit times at the current port speed. This value is specified via the mac_pause_quanta[15:0] in the Mac_Global_Mode register.

Interface Selection. Ports 44 through 47 support both SGMII and SerDes interfaces. During packet transmission, both interfaces are active. However, during packet reception, only one interface may be active at a time for any particular Ethernet MAC. The input_select[1:0] fields of the Mac_Mode_{0..4} registers are used to select which interface is to be utilized for packet reception.

Data Pipeline Flushing. The gmac_tx_flush_{0..9} bits in the Mac_Mode_{0..4} registers are used to flush any nonempty queues associated with a MAC that has been disabled from transmitting for whatever reason. While in the flush mode, packets are drawn out of the packet buffer in a normal manner and are then discarded prior to transmission. This has the effect of freeing up storage and queue resources that would have otherwise been stalled behind a disabled MAC.

Ideally, the forwarding characteristics of the device are updated to prevent further new queue entries for a disabled port prior to initiating a flush. The Layer_2_Global_Mask register may be used as a quick and effective way to disable new queue entries.

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

PHYs

MDIO. The external PHY devices are configured and controlled via an MDIO interface. This interface is accessed via the Mdio_Control, Mdio_Mode, and Mdio_Status registers.

The Mdio_Control register is used to issue read and write commands to the ET3028-50's MDIO controller. To issue a write command, the supervisor first sets up the parameters of the command (e.g., MDIO register address, port select, device select, etc.) by writing to Mdio_Control with an mdio_opcode[1:0] value of 00_2. Once the parameters have been established, the write data is then written to Mdio_Control and the command is automatically executed.

A read command starts off similarly: the supervisor sets up all of the various parameters with an opcode of 11_2. This opcode results in an immediate initiation of the read command. The mdio_busy bit remains asserted throughout the execution of the command. Once mdio_busy is deasserted by the ET3028-50, the read data is available in the mdio_addr_data[15:0] field.

A separate register, Mdio_Status, is available for polling the state of the MDIO controller. The mdio_busy bit in this register has the same meaning as the version of the bit that appears in the Mdio_Control register. The mdio_done bit is asserted at the completion of a command. This bit remains asserted until a one is written to its position within Mdio_Status by the supervisor.

The Mdio_Mode register is used to define the characteristics of the MDIO clock. The period of the clock is defined by the mdio_clk_period[7:0] field. Larger values of mdio_clk_period[7:0] result in lower MDIO clock frequencies. The mdio_clk_offset[7:0] is used to establish a delay between the MDIO clock and the transition or sampling of MDIO data.

SerDes. The four 1.25 GHz SerDes that make up the four 1 Gbit/s SFP interfaces require a certain amount of attention from the supervisor. Access to the control and status registers within the SerDes is available via the Serdes_Control_{4} register.

Both 8-bit and 16-bit access commands are available. These two command types share a common 32-bit register location. In other words, the format of the fields at this register offset depends on the type of command being executed. The target SerDes register address for the command and the write data (for write commands) are written to serdes_addr[15:0] (or serdes_addr[7:0]) and serdes_wr_data[15:0] (or serdes_wr_data[7:0]), respectively. Next, the access command code is written to the access_command[2:0] field at the same time that the command_start bit is asserted.

Setting the command_start bit initiates the specified command. The serdes_command_busy bit is asserted to indicate that the command is being executed. At the completion of the command, serdes_command_done is asserted to indicate that read data is valid (in the case of a read command) and that a subsequent command may be issued. The serdes_command_done bit remains asserted until the initiation of a subsequent command.

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

Link Aggregation

Link aggregation enables multiple, parallel Ethernet links to appear to the ET3028-50 as if they were a single, logical link; thus providing an increase in bandwidth. No more than eight ports may be a part of any one aggregate.

Logical Ports. Link aggregation enables the possibility of a single MAC source address appearing on several physical links nearly simultaneously. Ordinarily, the appearance of a MAC source address on a receive port that is different than the one listed in the address table triggers an address learning event. However, for the ports within an aggregate, all of the ports in that aggregate are valid receive ports for any particular source address. Hence, for address learning purposes, the concept of a logical port is used.

Physical ports are mapped to logic ports through the use of the `Layer_2_Logical_Port_Table` register. This table is addressed by the physical receive port number, and it returns a logical port number. For an aggregate of links, all of the physical receive ports that are part of the link should return a common logical port number that is equal to the lowest physical port number of the ports in the aggregate. For those ports that are not part of an aggregate, this table should be configured to return those ports' physical port number. Complying with the preceding recommendation is not essential. Alternate methods of configuring this table may be employed. The only important criteria is that all ports in an aggregate must share a common logical port number.

Link Selection. Packets received on any of the ports that make up a particular aggregate are treated identically by the ET3028-50. Transmit packets may be transmitted by any of the ports of an aggregate with the actual transmit port being selected arbitrarily. The only requirement for transmit port selection is that all packets that may be part of a particular conversation or flow use a common physical port.

The ET3028-50 uses an arithmetic reduction of the receive packet's MAC destination and source address values to arrive at a 3-bit link selection value; enabling a one-of-eight selection. How this one-of-eight selection is employed is described below.

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

Destination Maps. In support of link aggregates, the `Layer_2_Dest_Map_Table` must be configured appropriately. If an aggregate is the intended destination for a particular address table entry/destination map association, then all of the bits corresponding to the ports in the aggregate must be asserted in the destination map. In other words, the destination map must be set up as if the packet is to be multicast to all of the ports in the aggregate. Aggregate masks are subsequently used to select a single port per aggregate.

Aggregate Masks. A total of eight aggregate masks are available via `Layer_2_Aggregation_Mask_Table`. These eight masks serve all of the possible aggregates that may be configured in the system.

Every physical transmit port can be thought of as belonging to some form of an aggregate. Conceptually, even a single-port aggregate is supported. The pattern of mask bits programmed into the `Layer_2_Aggregation_Mask_Table` is determined by the number of ports in the aggregate. Membership levels of 1 through 8 ports are allowed, though 1, 2, 4, and 8 are likely to be the most commonly used. The following table shows the entries for aggregates of various port-counts.

Table 338. Aggregate Mask Patterns

Aggregate Mask Number	Mask Patterns for Various Ports Per Aggregate			
	1 Port	2 Ports	4 Ports	8 Ports
0	0_2	01_2	0111_2	01111111_2
1	0_2	10_2	1011_2	10111111_2
2	0_2	01_2	1101_2	11011111_2
3	0_2	10_2	1110_2	11101111_2
4	0_2	01_2	0111_2	11110111_2
5	0_2	10_2	1011_2	11111011_2
6	0_2	01_2	1101_2	11111101_2
7	0_2	10_2	1110_2	11111110_2

For a single-port aggregate (really, a port that is not a part of any aggregate), the bits that correspond to that port are deasserted (not masking) for all of the `Layer_2_Aggregation_Mask_Table` entries. In other words, regardless of which table entry is selected by the arithmetic reduction of the packet's MAC destination and source address values, the destination port is never masked by the aggregation mask table.

For an eight-port aggregate, each aggregation mask table entry enables just one of the eight ports (note the marching zero in the table above). Each mask bit in the 8-bit values shown above corresponds to one of the eight ports in the aggregate.

Note: Neither the destination maps nor the aggregation masks impose any limits about which ports may be designated to be a member of a particular aggregate. The ports in an aggregate need not be adjacent ports nor need they even be equivalent ports.

VLANs

Every receive packet is associated with one VLAN or another by the ET3028-50 during reception. There are several methods by which such a VLAN association may be made.

1. Directly. The VLAN tag of a received packet is mapped to a VLAN index.
2. Port/protocol. Untagged packets are mapped to a VLAN index based on a per-port protocol table.
3. Port only. Untagged packets adopt the receive port's default VLAN ID.

For the purposes of restricting the forwarding of packets within VLANs, a series of VLAN masks are used. The nature of these masks is described in detail further below.

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

Direct Mapping. Although the *IEEE 802.1q* standard allows up to 4,094 VLAN IDs, the ET3028-50 supports up to 256 active VLANs. The `Vlan_Index_Table_{0..6}` registers are used to map the 12-bit VLAN tag from a receive packet to the 8-bit VLAN identifier that is used internally within the ET3028-50. This table is addressed directly by the receive packet's 12-bit VLAN tag. A `vlan_index_valid` flag in each table entry identifies those entries that are indeed valid. For valid entries, the `vlan_index[7:0]` value from the same record is used as the receive packet's VLAN index.

If the receive packet's 12-bit VLAN ID maps to an invalid entry in `Vlan_Index_Table_{0..4}`, then the packet is either discarded or its VLAN index is chosen by one of the alternate methods listed above and described in more detail below. The `invalid_vlan_id_discard_en` bits in the `Port_Mode_{0..4}` registers are used to set this preference on a port-by-port basis.

Note: Revisions B1 and C of the ET3028-50 allow up to 4,095 VLAN IDs. VLAN ID 0xFFF is included for non-standard use.

Port/Protocol Mapping. For each receive port, as many as eight Ethertype values may be mapped to VLAN index values. The following ordered list of Ethertypes is used.

Table 339. Ethertype Index Values

Ethertype	Ethertype Index
IPv4	0
IPv6	1
ARP	2
RARP	3
User Type 0	4
User Type 1	5
User Type 2	6
Unknown	7

The three user type values are established via the `User_Type_{0..4}` registers. Typically, all five instances of this register are programmed the same way, but it is not necessary to do so.

The Ethertype index is used to select one of the eight VLAN index values from the `Vlan_Port_Protocol_Table_{0..4}` registers. There is an eight-entry table for each receive port.

Valid entries are identified by an asserted `entry_valid` bit. If the combination of receive port number and Ethertype index points to a valid entry, then the associated `vlan_index[7:0]` value is assigned to the receive packet. Otherwise, the port-only VLAN assignment method (described below) is used.

Port Only Mapping. Should the preceding methods for assigning a VLAN index to a receive packet fail, the receive port's default VLAN index is used. This value is established on a per-port basis via the `Port_Mode_{0..4}` registers.

VLAN Masks. 256 VLAN masks are available that are applied to the 38-bit destination map during the bridging process (see Destination Maps on page 247). These masks are configured via the `Layer_2_Vlan_Mask_Table` register. The receive packet's VLAN index is used to select the appropriate VLAN mask. The mask is then used to eliminate all destinations that are not members of the packet's VLAN.

Appendix B: Configuration (continued)

Ethernet Interfaces (continued)

Port Mirroring

To aid in the analysis of network behavior, ports may have their receive and transmit activity copied to a designated mirror port.

Designating a Mirror Port. A port is designated as the system's mirror port by writing its port number to the `Layer_2_Mirror_Port` register. When a receive packet is to be mirrored, the designated port is added to the 38-bit destination port map by the ET3028-50.

Designating a port as a mirror port does not automatically prevent that port from being used for normal forwarding purposes. In order to dedicate a port solely to mirroring, it is essential that the address table and/or destination port maps be updated to eliminate the mirror port as a valid destination.

If multiple ports are selected to be mirrored (i.e., their activity is copied to the designated mirror port), it is possible that the aggregate of the mirrored ports' bandwidth may exceed that of the mirror port. If this condition persists for an extended period, the packet buffer may become congested and packets may be discarded. In this case, the inclusion of a mirror port may adversely affect the behavior of the mirrored ports.

If both the transmit and receive ports for a particular packet are being mirrored, only one copy of the packet is sent to the mirror port.

Receive Mirroring. To mirror the receive activity of a port, it is merely required that its corresponding bit in `Layer_2_Src_Mirror_Map` be asserted. Once this is done, all packets received by the mirrored port are copied to the designated mirror port.

Transmit Mirroring. To mirror the transmit activity of a port, it is merely required that its corresponding bit in `Layer_2_Dest_Mirror_Map` be asserted. Once this is done, all packets transmitted by the mirrored port are copied to the designated mirror port.

Note: The packet transmitted by the mirror port may not exactly match the packet transmitted by a mirrored port if the VLAN modes and configurations of the two ports differ.

Bridging

Bridging refers to the forwarding of Ethernet packets through the use of Layer 2 information as defined by the *IEEE 802.1d* standard. Fundamental to the bridging process is the configuration of the Layer 2 address tables.

Layer 2 Look-Up Method

The ET3028-50 utilizes 4-ary search algorithms in its table look-up functions. The 4-ary search algorithm is implemented in a series of pipelined, cascaded key tables. Every search starts at `Layer_2_Key_Table_0` with the comparison of the search argument (one of the Layer 2 address fields from the receive packet) against the three keys stored in this register. These three keys have the effect of dividing up the overall table into four regions. Hence, the next stage of the search has the search argument being compared against one of four sets of three keys in `Layer_2_Key_Table_1`. This process continues until `Layer_2_Key_Table_7` is reached.

At this last stage, there is a pair of keys against which an exact match comparison is performed. If `layer_2_key_0[47:0]` matches the search argument, then the search results associated with key 0 are returned. If `layer_2_key_1[47:0]` matches the search argument, then the search results associated with key 1 are returned. If neither keys match the search argument, then a look-up failed indication is returned.

Appendix B: Configuration (continued)

Bridging (continued)

The associated data returned from a look-up operation consist of:

- A Layer 2 source address permit/deny indication.
- The receive port number associated with the matching address value when used as a source address.
- A 3-bit flow identifier for use when the matching address value is used as a source address.
- A 3-bit flow identifier for use when the matching address value is used as a destination address.
- A source log indication that causes the packet to be copied to the logging queue when the matching address value is used as a source address.
- A 9-bit destination map index that identifies one of 512 initial distribution patterns for the packet. This initial distribution pattern is subject to extensive modification during the remainder of bridging processing.

Adding and Deleting Table Entries

For the search algorithms to work properly, the keys must be placed in the tables in a numerically sorted manner. Consequently, if the two existing keys between which a new key must be added are not separated by one or more empty locations, then the table's contents must be shifted in order to open up such an empty location. This is done through a series of swaps of adjacent values. In other words, a form of bubble sort is performed.

For the sake of clarity, a binary table example is presented below. The methods described here are easily extended to the 4-ary case.

Swapping Binary Table Keys. Depending on which pair of table leaf nodes need to be swapped, the operation could involve the updating of nodes at every level of the tree's hierarchy. Consider the following figure.

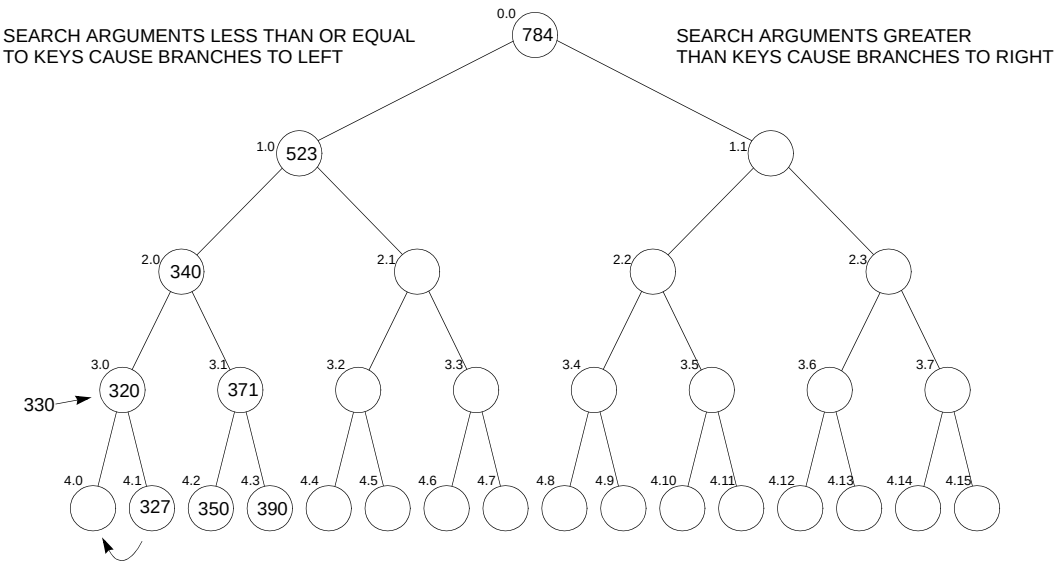


Figure 255. Updating a Binary Tree Structure (Step 1)

Appendix B: Configuration (continued)

Bridging (continued)

Since it is only permissible to swap occupied leaf nodes with empty leaf nodes, it is easy to see how swapping two leaf nodes that share a common immediate ancestor would be quite simple.

For example, presume that node 4.0 is empty and 4.1 contains a valid key, say 327. The first step is to copy the key in 4.1 to node 4.0. Since 3.0 hasn't been updated yet, its current value of, say 320, continues to direct searches looking for 327 to node 4.1. The search could, however, now obtain the same results by branching to 4.0. Therefore, it is safe to update node 3.0 to, say 330, so that search arguments of 327 result in a branch to 4.0. At this point, node 4.1 may be marked as invalid.

The updated structure is shown in the following figure.

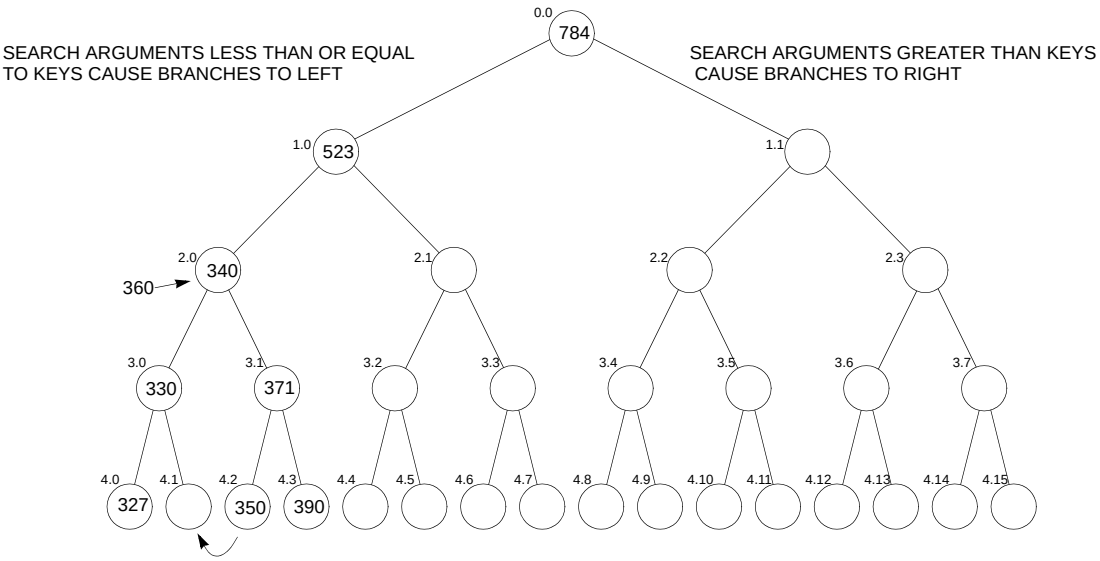


Figure 256. Updating a Binary Tree Structure (Step 2)

Continuing with the example, presume that we now want to swap nodes 4.1 and 4.2. Since node 4.1 is empty, the contents of node 4.2 may be freely copied into node 4.1. Currently, a search argument value of 350 causes a branch to the right at node 2.0. By depositing a value greater than 350, say 360, into node 2.0, future searches for 350 branch left at node 2.0 and then branch right at node 3.0. Node 4.2 may then be marked as invalid.

The binary tree examples presented above may be extended to 4-ary trees by bearing in mind that each node contains three keys so as to provide for a 4-way branch. This means that node n.0 depends on the first key in the parent node; node n.1 depends on the first and second key; node n.2 on the second and third key; while node n.3 depends on just the third key. Therefore, when nodes are being updated, it is important that all of the keys that are associated with a branch in their direction be properly updated as well.

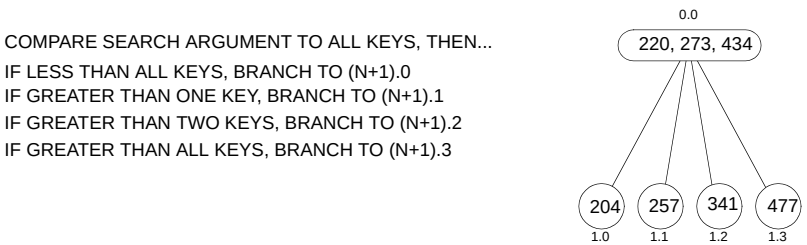


Figure 257. Updating a 4-ary Tree Structure

Appendix B: Configuration (continued)

Bridging (continued)

Populating Upper Levels of Tree Hierarchy. Since the ET3028-50 uses physically distinct memory systems for each level of hierarchy in its address table tree structures, it is not necessary to use actual address table key values in the upper levels of the tree structure. All that matters is that a value is in place that causes a comparison to result in a branch in the correct direction. Consider the example in the following figure.

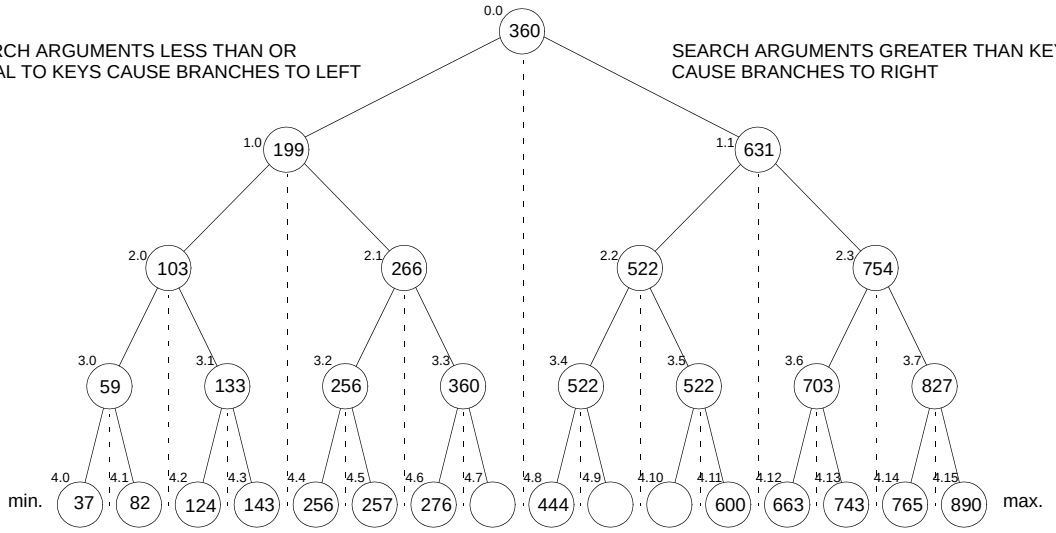


Figure 258. Populating Upper Levels of a Binary Tree Structure

For this example, the leaf nodes have been populated with a sorted list of arbitrary values. Three of the leaf nodes have been left empty. The values in the upper nodes are simply integer averages of the two leaf nodes directly below. These leaf nodes are identified by the dotted lines descending from the upper nodes. The two leaf node keys to use to form the average are the two immediately to the left and right of the dotted line. In the case of an empty leaf node, continue to scan left or right (as is appropriate) until reaching a nonempty leaf node. In the example, node 2.2 used nodes 4.8 and 4.11 for the key values to average (as did nodes 3.4 and 3.5). If the end of the table is reached before encountering a nonempty node, then a minimum or maximum constant is used instead, as is appropriate.

Note that the only instance where an actual leaf key value is used in an upper-level node is for node 3.2. This is so because its two leaf nodes are consecutive and an integer average is guaranteed to be equal to the lesser of the two values. Even though an exact match with a search argument may occur in node 3.2, the searching continues until node 4.4.

If the value 300 were deposited into node 4.7, then node 3.3 would be updated to (276 + 300)/2 = 288. This will cause the appropriate branching to occur. For the sake of consistency, node 0.0 would also be updated. Its new value would be (300 + 444)/2 = 372. Note that making this change to node 0.0 does not affect table behavior.

However, if node 4.7 were updated to 400, then changing node 3.3 to (276 + 400)/2 = 338 would have no effect but the changing of node 0.0 to (444 + 600)/2 = 522 causes the new leaf node to be found properly.

Now consider the swapping of node 4.7 and 4.8. This requires only the changing of node 0.0 to (444 + 600)/2 = 522, which then results in the correct behavior of the table.

Appendix B: Configuration (continued)

Bridging (continued)

Identifying Upper Nodes. The method for identifying which upper node is associated with any particular pair of adjacent leaf nodes is to first identify the appropriate level of hierarchy and then the node within that level. The hierarchy level is identified by taking the exclusive OR of the index values of the two leaf nodes under consideration and then counting the number of ones in the result. The number of ones indicates the number of levels that much be traversed from the leaf node level.

For example, the upper node associated with nodes 4.9 and 4.10 is identified by the XOR of 9 (1001_2) and 10 (1010_2) = 0011_2. The two asserted bits means that the upper-level node is two levels above the leaf node level (level 2). To identify the index at that level, use the smaller of the two leaf node index values (9, in this example) and shift it to the right by the number of levels that must be traversed upwards (2, in this example). $1001_2 >> 2 = 10_2 = 2$. Hence, the upper-level node associated with leaf nodes 4.9 and 4.10 is node 2.2.

Address Aging

An address aging table (`Layer_2_Time_Stamp_Table`) is used to keep track of the time that has elapsed since the observation of each MAC source address. Source addresses that have not been seen in a receive packet for a certain period of time may be deleted from the address table by the supervisor.

When a Layer 2 source address matches an entry in the address table, its corresponding entry in `Layer_2_Time_Stamp_Table` is updated with the current value of `Layer_2_Current_Time`.

Periodically, the supervisor scans through the time stamp table, looking for entries that are equal to the next value of `Layer_2_Current_Time`. These entries correspond to address table entries that are set to expire upon the next incrementing of `Layer_2_Current_Time`. Once the supervisor has deleted these entries, it may increment `Layer_2_Current_Time`.

Note: There is a one-to-one correlation between address table entries and aging table entries. As entries in the address table are swapped to allow for the sorting of new entries, the same swap operations must be performed on the time stamp table.

Destination Maps

A 38-bit wide vector is used to specify the destinations for forwarded packets. The least significant (rightmost) 28 bits each correspond to one of the ET3028-50's Ethernet ports.

The most significant 8 bits (bits 50 through 57) of the 38-bit vector correspond to the eight queues that are dedicated to the supervisor. Each of these queues may be dedicated to a particular purpose (e.g., address learning, logging, BPDU reception, etc.).

Bits 48 and 49 of the destination map are unused.

The initial destination map is retrieved from the `Layer_2_Dest_Map_Table` register. This table is addressed by the `dest_map_index_{0..1}[7:0]` value retrieved from the `Layer_2_Key_Table_6` record that contains a key that matches the destination address of the receive packet.

For unicast packets (i.e., a single destination port), only one bit in the destination map is asserted. For multicast packets, multiple bits may be asserted. Each asserted bit represents an intended destination.

Certain subsequent filtering and processing steps that follow the establishment of the initial destination map may serve to add or delete (assert or deassert, respectively) destinations to or from the bit map.

Appendix B: Configuration (continued)

Bridging (continued)

Packet Flooding

According to the *IEEE 802.1d* standard, a packet whose destination address is unknown (i.e., not present in the Layer 2 address table) must be flooded to all potential destination ports. If a destination address look-up results in a look-up failure, then the port map contained in `Layer_2_Flood_Map` is used as the initial destination port map rather than a value from `Layer_2_Dest_Map_Table`.

The flood map is subject to pruning by such functions as VLANs.

Address Learning

If the look-up on the receive packet's source address indicates that the value is not present in the Layer 2 address table, then the learning of the source address value is called for. The supervisor is responsible for carrying out the actual addition of the address value to the Layer 2 address table. The ET3028-50 merely recognizes that a particular source address is unknown, and forwards a copy of the packet to the supervisor, and then allows the supervisor to determine where (if anywhere) in the table the entry belongs.

The supervisor queue designated to be the learning queue is specified via the `Layer_2_Learning_Port` register. Ports 50 through 57 correspond to the supervisor's eight receive queues.

Note: Each supervisor queue may be independently configured to truncate receive packets. This feature may be beneficial for the address learning function if packet data beyond the MAC header is of no interest to the supervisor.

Layer 2 Access Control

Each Layer 2 address table entry includes an access control flag bit as part of an address value's associated data. This flag bit acts as an access control indicator. The flag is examined when the address being submitted for a look-up is a source address. Namely, this flag is the `layer_2_src_permit_{0..1}` bits of the `Layer_2_Key_Table_6` register. There is one flag for key 0 and one for key 1 in each record.

If the flag bit for a particular source address is deasserted, the affected receive packet is denied access to the device. This denial does not necessarily imply that the packet is discarded. Rather, the mask stored in `Layer_2_Src_Deny_Mask` is applied to the destination map for the packet. Certain destinations such as supervisor queues may remain enabled for logging or other purposes.

Access control based on a packet's destination address is simply a matter of using a `layer_2_dest_map_index_{0..1}[7:0]` value in `Layer_2_Key_Table_6` that points to a destination map whose pattern of asserted bits is appropriate for a Layer 2 denial (e.g., all bits deasserted).

Spanning Tree States

The *IEEE 802.1d* spanning tree protocol requires that the various ports transition through a series of states; each state implying a certain per-port behavior. These states are listed in the following table.

Table 340. Spanning Tree states

State	Receive?	Learn?	Forward?
Disabled	No	No	No
Blocking	Yes	No	No
Listening	Yes	No	No
Learning	Yes	Yes	No
Forwarding	Yes	Yes	Yes

Note: Operationally, there is no real difference between the blocking and listening states. Hence, they are combined into a single state called blocking for the purposes of explaining how to configure the ET3028-50.

Appendix B: Configuration (continued)

Bridging (continued)

Disabled State. In this state, all reception is disabled. To accomplish this for all receive traffic on a particular port, the simplest method is to deassert the `gmac_rx_en` or `xgmac_rx_en` bit in the appropriate `Mac_Mode_{0..4}` register. Transmissions via a particular port are easily disabled by configuring `Layer_2_Global_Mask` to eliminate that port from the destination map.

To disable all reception on a particular port, it is a matter of setting the `stp_state[1:0]` value that corresponds to the port of interest in the `Layer_2_Port_State_Table` register. When a port is in the disabled state, then the `Layer_2_Src_Deny_Mask` register's value is applied to the packet's destination map by the Layer 2 look-up function. The code for the spanning tree port disabled state is 11_2.

Blocking State. In the blocking state, BPDU packets may be forwarded to the supervisor, but all other receive packets are discarded and no packets other than BPDUs are transmitted via the port. A port is placed in the blocking state by making an appropriate entry in the `Layer_2_Port_State_Table` register. Each entry in the table is a 2-bit spanning tree state code. The code for the blocking state is 00_2.

During the bridging process, the state of the port is tested. If the blocking state is detected, then the `Layer_2_Blocking_Mask` is applied to the destination bit map. This mask is typically configured to eliminate all destinations except for the supervisor queue that has been designated to receive all BPDU packets. Hence, all non-BPDU packets are discarded. Other exceptions to the application of this mask may be configured as desired.

The prevention of transmissions by a blocked port is achieved through the proper configuration of the various VLAN masks, essentially eliminating the port from all VLANs.

Learning State. In the learning state, unknown source addresses are expected to be added to the Layer 2 address table. However, the general forwarding of packets is not permitted. A port is placed in the learning state by making an appropriate entry in the `Layer_2_Port_State_Table` register. Each entry in the table is a 2-bit spanning tree state code. The code for the learning state is 01_2.

During the bridging process, the state of the port is tested. If the learning state is detected, then the `Layer_2_Learning_Mask` is applied to the destination bit map. This mask is typically configured to eliminate all destinations except for the supervisor queues that have been designated to receive all BPDU packets and those packets with unknown source addresses. Hence, all other packets are discarded. Other exceptions to the application of this mask may be configured as desired.

Transmissions by a port that is in the learning state is disallowed and is achieved through the proper configuration of the various VLAN masks, essentially eliminating the port from all VLANs.

Forwarding State. In the forwarding state, packets are forwarded normally. A port is placed in the forwarding state by making an appropriate entry in the `Layer_2_Port_State_Table` register. Each entry in the table is a 2-bit spanning tree state code. The code for the forwarding state is 10_2.

Appendix B: Configuration (continued)

Access Control Lists

Access control lists (ACLs) provide a means for restricting which receive packets are permitted access to the ET3028-50 and which are not. Each port or VLAN is associated with a single ACL. Each ACL consists of a list of access control entries (ACEs). Each ACE consists of a 5-tuple of values that are compared against various fields from the receive packet. These fields are:

1. IPv4/IPv6 source address
2. IPv4/IPv6 destination address
3. Ethertype/protocol
4. TCP/UDP source port
5. TCP/UDP destination port

The comparisons are used to identify the first ACE in an ACL that matches all of the criteria. The action associated with that ACE are then carried out. The possible actions include:

1. Permit
2. Deny
3. Log
4. Adjust Priority

Enabling ACLs

ACLs are individually enabled or disabled by asserting or deasserting the bits in `Acl_En`. An ACL must be enabled in order for it to potentially apply the deny action to a receive packet. A disabled ACL (its corresponding bit in `Acl_En` deasserted) implies an automatic permit action to all receive packets mapped to that ACL.

ACL Modes

IPv4 vs. IPv6. Various basic operating modes of the ACL function are established by `Acl_Mode`. The `ipv4_only` bit being asserted causes the IP address tables to be restricted to just 32-bit IPv4 addresses. Doing so sets the table's capacity to 64 IP subnets or host addresses.

Deasserting the `ipv4_only` bit enables a mix of IPv4 and IPv6 addresses. The trade-off is that the IP address table's capacity is reduced to 16 IP subnets or host addresses.

Port- vs. VLAN-Based. When the `port_based_acls` bit is asserted, each port is associated with a single ACL. The receive port number corresponds to the ACL number.

Deasserting the `port_based_acls` bit causes the receive packet's VLAN to be used to associate the packet with a VLAN. Since there are potentially more VLANs (256) than available ACLs (48), a method is provided to indirectly map between the two. This mapping is performed by the `Acl_Vlan_Index_Table` register.

Logging Denied Packets. Ordinarily, the logging (copying to the supervisor) and denial of a packet are independent actions. However, it may be desired to log those packets that were denied by default (matched no ACEs). Asserting the `auto_deny_log_en` bit in the `Acl_Mode` register causes all packets denied by default to be copied to the supervisor.

The supervisor queue that is dedicated to ACL logging is chosen by `Packet_Buffer_Acl_Log_Port`.

Appendix B: Configuration (continued)

Access Control Lists (continued)

IP and TCP Tables

The IP and TCP tables utilize a cascaded *k*-ary look-up method. A series of 2-way or 4-way decisions are made via a series of tables arranged as a balanced tree structure.

All *k*-ary tables have a look-up entry point (table 0) and a look-up terminus point. The look-up terminus point varies from table to table and is affected by the type of look-up (binary vs. 4-ary) and the number of keys in the table.

The IP address table operates in either a binary or 4-ary mode, depending on whether it is a mix of IPv4 and IPv6 addresses or exclusively IPv4 addresses being operated on, respectively. For a mix of IPv4 and IPv6, there are a total of nine stages to the look-up process with the final stage holding 16 keys defining 15 prefixes. Hence, this type of look-up starts with `Acl_Ip_Key_Table_0` and ends with `Acl_Ip_Key_Table_8`.

For IPv4-only operations, there are a total of six look-up stages with the final stage holding 64 keys that define 63 prefixes. Hence, this type of look-up starts with `Acl_Ip_Key_Table_0` and ends with `Acl_Ip_Key_Table_5`.

The TCP/UDP port number look-ups operate in a 4-ary mode and require five stages with the final stage holding 16 keys that define 15 ranges. The look-up entry point is `Acl_Tcp_Key_Table_0` and the terminus is `Acl_Tcp_Key_Table_4`.

From one look-up stage to the next, there are no pointers embedded in the table that point to the keys to be examined within the next table in sequence. Rather, it is the progression of match results that forms the address used to pull the appropriate keys from the next table. Using a 4-ary look-up as an example: At each stage, the address of the current set of keys is shifted left two bit positions and the results of the current comparisons serve as the new least significant two bits of the address for the next key table. The following table presents an illustration.

Table 341. Key Address Formation

Comparison Results	New Address Bits
argument <= key_0	00_2
argument > key_0 and <= key_1	01_2
argument > key_1 and <= key_2	10_2
else	11_2

The records embodied in the last stage of each of the look-up types contain both key values as well as look-up results. These look-up results are index values that identify which prefix or range (if any) matches the search argument. For the IP address look-ups, source and destination address look-ups return distinct index values for source vs. destination address look-ups as well as flow identifier values that are also distinct for source vs. destination address look-ups.

For details regarding the method used to add and delete table entries, refer to Adding and Deleting Table Entries on page 244.

Appendix B: Configuration (continued)

Access Control Lists (continued)

Index Tables

A series of tables are used to establish indirections from one stage of ACL processing to the next. These indirections minimize redundancy and provide the opportunity for the sharing of table entries from one ACE to the next. The following figure shows how the various index tables are incorporated into the look-up process.

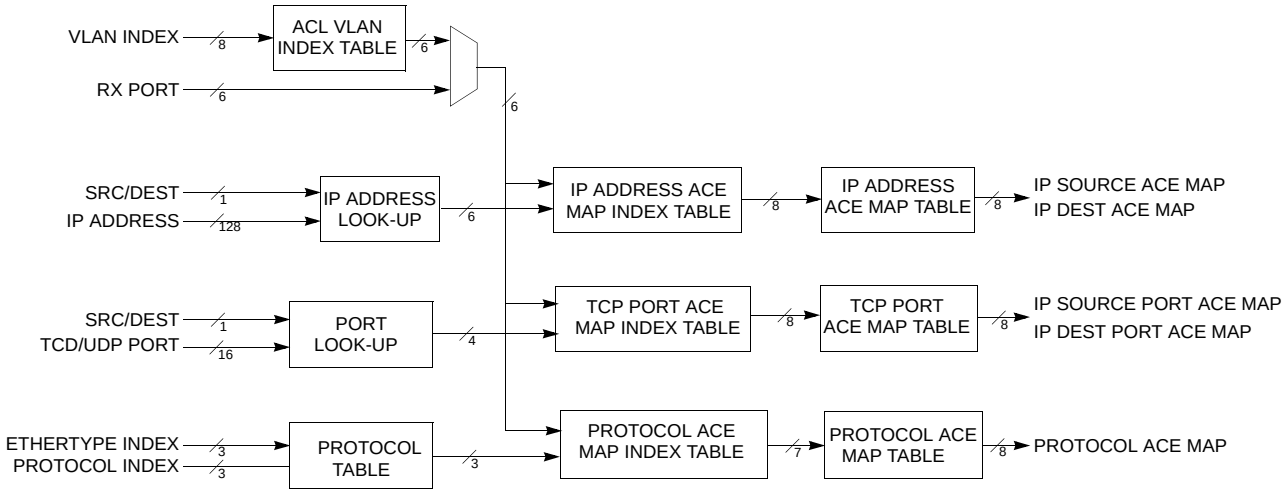


Figure 259. ACL Processing Structure

VLAN Index Table. When the port-based ACL mode is selected via `Acl_Mode`, there is a simple one-to-one association between a receive port and its associated ACL. Essentially, the port number and the ACL number are the same.

When the ACL function is configured to be VLAN-based, `Acl_Vlan_Index_Table` is used to establish the mapping between a VLAN index and an ACL. This register is a 256-entry table that is addressed by the receive packet's 8-bit VLAN index. Returned from this table is a 6-bit value (`acl_index[5:0]`) that is used to select an individual ACL.

Note: There is an important distinction between a VLAN identifier and a VLAN index. A VLAN identifier is the 12-bit value received in a VLAN tagged packet. A VLAN index is the 8-bit value used to represent the VLAN association of the packet within the ET3028-50 device.

The following figure illustrates the relationship between the VLAN identifier, VLAN index and ACL.

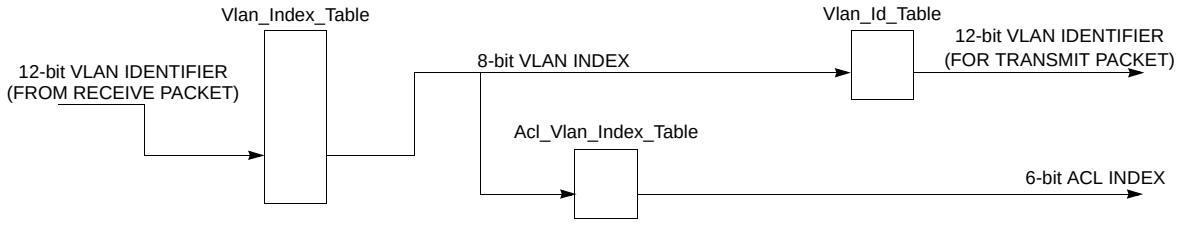


Figure 260. VLAN-to-ACL Mapping

Appendix B: Configuration (continued)

Access Control Lists (continued)

IP Address ACE Map Index Table. This table is used to define the relationship between the IP address index (look-up result), ACL number, and ACE map. The IP address index identifies one of 64 address values. This means that there can be, at most, 64 unique IP address look-up results. These results must be shared by all of the ACLs. The 6-bit look-up result is concatenated with the output of `Acl_Vlan_Index_Table` to form a 12-bit address that is then delivered to `Acl_Ip_Addr_Ace_Map_Index_Table`.

The output of `Acl_Ip_Addr_Ace_Map_Index_Table` is a 10-bit value that is used to select one of 1,024 IP address ACE maps. These ACE maps define the patterns of appearance of particular IP address prefixes or host addresses within ACEs. If multiple ACLs happen to share the same pattern of appearance for a particular IP address value, then this table can be used to point the multiple IP address look-up results to a common ACE map.

TCP Port ACE Map Index Table. The 4-bit results of the TCP/UDP port look-up are concatenated with the 6-bit ACL index to form a 10-bit address used by `Acl_Protocol_Ace_Map_Index_Table` to retrieve one of 768 8-bit ACE map index values.

This index table enables the reuse of ACE maps by the TCP/UDP look-up results. Say, for example, that the TCP destination port range of 100 through 150 appears in ACE number 4 for five different ACLs. It is then a simple matter of having that particular ACE map be pointed to by the five locations in the index table addressed by the appropriate combinations of look-up results and ACL index.

Protocol ACE Map Index Table. Up to eight Ethertypes and eight Layer 4 protocols may be grouped into a maximum of eight combinations. This combined Ethertype/protocol index is used to identify the protocol value for this field of the ACE 5-tuple. The `Acl_Protocol_Ace_Map_Index_Table` is used to map the 512 protocol/ACL number combinations to 128 ACE maps.

ACE Maps

ACE maps are used to identify those ACEs that have criteria that match the corresponding 5-tuple field from the receive packet. For example, if the receive packet's IP destination address matches the criteria for ACEs 1, 2, and 7 for ACL 18, then that IP address look-up/ACL number combination is mapped (via `Acl_Ip_Addr_Ace_Map_Index_Table`) to an ACE map wherein bits 1, 2, and 7 are all asserted and the remaining 5 bits are deasserted. This process of selecting ACE maps is repeated for all fields of the 5-tuple. The result is a series of five 8-bit ACE maps, each indicating the ACEs whose criteria are satisfied by the fields of the packet. It is then a simple matter of finding the first (lowest numbered) ACE for which all five ACE maps have the corresponding bit asserted. The following figure illustrates this process.

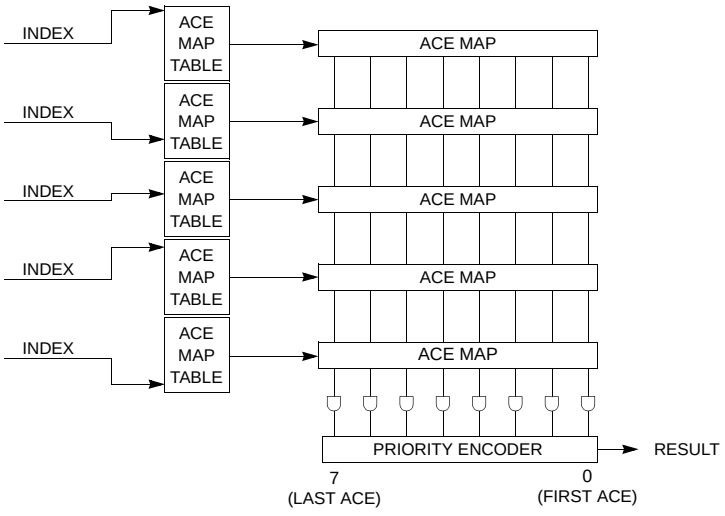


Figure 261. ACE Map Reduction

Appendix B: Configuration (continued)

Access Control Lists (continued)

The ACE maps are defined via `Acl_Ip_Addr_Ace_Map_Table`, `Acl_Port_Ace_Map_Table` and `Acl_Protocol_Ace_Map_Table`.

ACL Actions

Associated with each of the access control entries is a set of actions to be carried out when a receive packet matches a particular ACE. These actions are as follows:

1. Permit or deny access.
2. Copy packet to logging queue.
3. Replace packet's priority.

These actions are specified by the `acl_permit`, `acl_log`, and `acl_priority_code[4:0]` fields of `Acl_Result_Table`.

Permit/Deny. All receive packets are associated with one ACL or another. This association is based on either the packet's receive port or its VLAN index. If the associated ACL is empty (i.e., no ACEs), then an automatic permit is implied. If a receive packet fails to match all ACEs in a nonempty ACL, then an automatic deny is implied. Finally, the first matching ACE's `acl_permit` bit is used to either permit or deny the packet's access to the ET3028-50 device.

If the `acl_permit` bit is set, the packet is received and processed normally. If this bit is deasserted, then the matching packet is denied access. This does not necessarily imply that the packet is discarded. It may still be forwarded to certain destinations such as the supervisor or a mirroring port. The effect of an ACL denial is specified by `Packet_Buffer_Acl_Deny_Mask`.

ACL Logging. ACL logging refers to the copying of certain packets to the supervisor that match an ACE whose `acl_log` bit is asserted. In general, logging only occurs when explicitly configured to do so, i.e., when an ACE is encountered with `acl_log` asserted. The automatic logging of packets denied by defaults (matched no ACEs) may also be enabled by setting `auto_deny_log_en` in the `Acl_Mode` register. The logging port or queue is identified by the value set in `Packet_Buffer_Acl_Log_Port`.

Priority Replacement. One of the functions available via access control lists is the ability to override the priority value of a receive packet based on a matching ACE. Every ACE result includes an `acl_priority_code[4:0]` field. If bit 4 of this field is asserted, bits 3:0 are used to overwrite the priority value that had previously been assigned to the receive packet during ingress parsing.

Appendix B: Configuration (continued)

Quality of Service

The ET3028-50 supports both ingress and egress quality of service (QoS) features. In the ingress direction, these features are related to rate limiting policers and queue assignments. In the egress direction, these features are related to rate limiting traffic shapers, queue prioritization, and queue service scheduling.

Ingress Policers

The ingress policers have the ability to limit the receive bandwidth of designated traffic flows. Since there is no appreciable buffering upstream of the policers, they affect their rate limiting by discarding packets as necessary (actually, a discard mask is applied to the packet's destination map).

Identifying Flows. Flows are identified by a 3-tuple that consists of a packet's Layer 2 destination address, source address and its priority (Layer 2 or 3).

There is a pair of 3-bit flow ID values associated with every MAC address table entry; one flow ID for an address value as a source and one for it as a destination. These values are established via the `layer_2_src_flow_id[2:0]` and `layer_2_dest_flow_id[2:0]` fields of `Layer_2_Key_Table_6`.

The priority value of a packet can be derived from a packet's receive port, the Layer 2 CoS field from its *IEEE 802.1p* tag or from the DSCP field of its IP header.

For Layer 2/CoS, the priority value is mapped directly from the CoS field of the packet (zero extended from 3 bits to 4 bits). If the CoS field is not present, the port's default priority is used.

For the DSCP field, it is mapped via a look-up table from a 6-bit value to a 4-bit value via `Priority_Encode_Table_{0..4}`.

Layer 3 priority information is ordinarily used for all IP packets while Layer 2 information is used for all non-IP packets. The device can be configured to always use Layer 2 priority information by asserting the `layer_2_priority_override_en` bit in the `Port_Mode` register.

The combining of the source and destination-based flow identifiers and the packet's priority is accomplished by `Policer_Flow_Id_Table{0..4}`. These tables are addressed by a concatenation of the receive packet's destination flow identifier (3 bits), source flow identifier (3 bits), and its priority value (4 bits).

There are eight policers associated with each of the 28 Ethernet receive ports. Hence, `Policer_Flow_Id_Table{0..4}` returns a 3-bit value.

Establishing Flow Bandwidth Limits. Two parameters are used to establish bandwidth limits on a per-policer basis: the policer delta and limit. These are established in `Policer_Delta_Table` and `Policer_Limit_Table`, respectively.

The policer delta is subtracted from its associated policer accumulator (`Policer_Accumulator_Table`) on a periodic basis. The accumulator will not go negative. The period used to apply the delta is proportional to the configured bandwidth for the port. In other words, the delta is applied 10 times more often for a 1 Gbit/s Ethernet port than it is for a 100 Mbits/s Ethernet port. The port speed for policer purposes is set in the `port_speed[1:0]` field in the `Policer_Accumulator_Table{0..4}` registers. Smaller values of this field correspond to smaller delta periods and, hence, higher port speeds.

Appendix B: Configuration (continued)

Quality of Service (continued)

Establishing Flow Bandwidth Limits (continued). Each received packet causes its associated policer accumulator to be incremented by a value that equals the length of the receive packet (as measured in bytes). If the receive byte rate exceeds the rate at which the deltas are being applied, then the value in the accumulator will tend to increase over time. Hence, an increasing accumulator value equates to an out-of-profile flow of traffic.

The limit established for a policer defines the maximum duration of out-of-profile traffic that is tolerated without penalty. In other words, a flow may be allowed to operate in a bursty manner as long as the aggregate data rate for the flow is below the data rate implied by delta value. The value established in `Policer_Limit_Table` defines the maximum tolerated burst or, to put it another way, the promptness with which the policer responds to out-of-profile traffic. A higher value in `Policer_Limit_Table` equates to longer bursts being tolerated or a somewhat slower response by the policer.

Specifying Actions for Out-of-Profile Traffic. Once a flow has been identified as being out of profile, the policer may take action on the packets associated with the flow. These actions are defined by the `Policer_Flow_Mode_Table` registers.

Each policer can be configured to do one of the following:

1. Ignore out-of-profile traffic
2. Discard all out-of-profile traffic
3. Discard all out-of-profile traffic only when the packet buffer is congested

Additionally, each policer can be configured to demote the priority of all out-of-profile traffic that is not discarded.

Egress Traffic Shapers

The egress traffic shapers are conceptually similar to the ingress policers in that they serve to limit the bandwidth allowed by each port. They differ, though, in that they don't discard excess traffic but, instead, merely retard it. This is possible because the packet buffer is upstream of the traffic shapers and can absorb sizable bursts of traffic whereas the policers have no such upstream buffering.

There is one traffic shaper associated with each transmit queue (recall that there are eight such queues for every transmit port).

The traffic shapers are credit-based mechanisms. On a periodic basis, the delta value programmed into `Packet_Buffer_Shaper_Delta` is added to a queue's shaper accumulator (`Packet_Buffer_Shaper_Accumulator`). The period for this accumulation is dependent upon the speed of the associated port (`Packet_Buffer_Port_Speed`). Higher port speeds result in proportionally higher delta accumulation rates.

The accumulation of these credits occurs continuously but ceases once the limit established in `Packet_Buffer_Shaper_Limit` is reached. The higher the limit, the longer the transmit burst that may occur.

Whenever the value of a traffic shaper's accumulator is greater than zero, the associated queue is enabled to arbitrate for transmission. Each successful transmission arbitration causes the byte count of the transmitted packet to be subtracted from the shaper's accumulator. If the accumulator's value goes negative, the queue becomes disabled from arbitrating until sufficient further credit has been accumulated.

Appendix B: Configuration (continued)

Quality of Service (continued)

Packet Priority Management

A packet's priority level is a critical characteristic that is taken into account throughout the forwarding process.

Priority Assignment. Upon reception, each packet is assigned a priority value. The most basic priority assignment comes from the receive port's default priority. This value is established via the `default_priority[3:0]` field of the `Port_Mode_{0..4}` registers. This default value is used whenever Layer 2 (VLAN) and Layer 3 (IP) priority information is not available. The `default_priority[3:0]` must be set such that lowest possible priority is represented by the value 0 while the highest priority is represented by the value 15.

VLAN-tagged, non-IP packets utilize the *802.1p* priority field directly (zero-extended to 4 bits).

IP packets utilize the DSCP field after having been mapped to 4 bits by the `Priority_Encode_Table` registers. These 64-entry tables are addressed by the 6-bit DSCP field from the IP header. The output of the table is the 4-bit priority value to be assigned to the packet. This table may be configured arbitrarily. However, it is recommended that lower-value entries relate to lower-priority levels.

Finally, if both Layer 2 (VLAN) and Layer 3 (IP) priority information is available in the receive packet, the Layer 3 information takes precedence unless `layer_2_priority_override_en` in `Port_Mode_{0..4}` is asserted.

Mapping to Queues. The ET3028-50 supports 16 levels of internal priority representation (as allowed by the 4-bit priority value assigned to each receive packet). However, there are eight queues available for each transmit port. Hence, the 16 packet priority levels must be mapped to the eight queues. This is accomplished via the appropriate programming of the `Packet_Buffer_Priority_Table` register.

This 16-entry table is addressed by the 4-bit priority value associated with each receive packet. The 3-bit output from this table is used to select one of the eight queues associated with each transmit port. Queue number zero is the lowest priority queue, while queue number seven is the highest.

Priority Remarking. A packet's embedded priority information may be changed prior to transmission if the ingress policer identified the packet as being out of profile and has moved to demote its priority.

A 32-entry table (`Priority_Decode_Table`) is used to establish new priority values for transmit packets. This table is addressed by a concatenation of a demote flag and the 4-bit priority value associated with the packet. The demote flag serves as the most significant bit of the table's address. Hence, the table can be thought of as having two halves. The lower half is the normal priority mapping while the upper half (addressed by an asserted demote flag) represents the demoted or lowered priority values for the transmit packet.

This table returns both a 6-bit DSCP value for the IP header as well as a 3-bit CoS field for the VLAN tag. These return values are used as is appropriate depending on the form of the transmit packet.

Appendix B: Configuration (continued)

Quality of Service (continued)

Packet Buffer Congestion Management

The central packet buffer is a limited resource that must be shared among all users (“users” being defined as receive ports). Various congestion thresholds are used to limit the number of buffers that may be allocated to each user.

Two forms of ingress flow control are available: policer and MAC. The policers may be configured to discard out-of-profile traffic at all times or only when the packet buffer is congested. Hence, the policers may be made responsive to the congestion state of the interface. The policers are configured for the type of action to take during congestion via the `Policer_Flow_Mode_Table` register. Setting the `discard_out_of_profile_when_congested` bit makes the respective policer responsive to the congestion state.

The MAC flow control utilizes *IEEE 802.3x* flow control packets to prevent the other end of a full-duplex Ethernet link from transmitting for a limited period of time. This form of flow control is not flow based and has the effect of shutting down all receive traffic on a port regardless of its priority or destination.

The congestion thresholds for these two flow control mechanisms are set via the `Packet_Buffer_Global_Congestion_Threshold` and `Packet_Buffer_Channel_Congestion_Threshold` registers. As the name implies, the global threshold is applied globally (affects all ports or channels) and is sensitive to the total allocation of buffers. The channel (or port) congestion threshold is applied to just a single group of ports and is sensitive to the number of buffers allocated to that receive group.

Each type of threshold (global or per-port) has two subtypes: policer or MAC.

Whenever the total number of buffers allocated exceeds a threshold set in `Packet_Buffer_Global_Congestion_Threshold`, the appropriate congestion indication (policer or MAC) is provided to all ports simultaneously.

If the global congestion threshold has not been exceeded, then when a per-port congestion threshold is exceeded, the appropriate indication (policer or MAC) is provided to only the affected port.

The congestion thresholds are typically set as follows:

- The policer thresholds should be lower than the MAC thresholds.
- The sum of the per-channel thresholds should be greater than the global threshold.

Queue Management (Revision C Only). In revision C, some new registers are added to create maximum queue sizes. To enable these new registers, `hol_mode` in the `Packet_Buffer_Mode` register must be asserted.

To create the maximum queue size, two sets of queue limits are configured. One set limits the number of packets that may occupy a queue during packet buffer congestion, and the second set limits the number of buffers (128 bytes) that may be occupied by each queue during packet buffer congestion. Each set of limits has a corresponding congestion threshold used to determine when the packet buffer is considered congested for these limits.

The `supervisor_queue_limit[12:0]` and `queue_limit[12:0]` fields in the `Packet_Buffer_Queue_Limit` register are global limits used to restrict the number of packets that may occupy a queue during packet buffer congestion. For these limits, the packet buffer is considered congested if `queue_mask_congestion_threshold[14:4]` in the `Packet_Buffer_Queue_Management_Thresholds` register is exceeded.

When this global threshold is reached and a queue is over its allocated packet limit, the ET3028-50 prevents further enqueues to that queue until the packet buffer is no longer congested or until the number of packets occupying the queue drops below its allocated limit. The packet buffer is no longer considered congested for these limits when the total number of queue entries drops below `queue_mask_congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4].`

Appendix B: Configuration (continued)

Quality of Service (continued)

Likewise, the `supervisor_queue_buffer_limit[14:0]` and `queue_buffer_limit[14:0]` fields in the `Packet_Buffer_Queue_Buffer_Limit` register are global limits used to restrict the number of buffers (128 bytes) that a queue may occupy during packet buffer congestion. For these limits, the packet buffer is considered congested, if `buffer_queue_global_congestion_threshold[14:4]` in the `Packet_Buffer_Queue_Management_Thresholds register` is exceeded.

When this global threshold is reached and a queue is occupying more buffers than its allocated limit, the ET3028-50 prevents further enqueues to that queue until the packet buffer is no longer congested or until the number of buffers occupied by the queue goes below its allocated limit. The packet buffer is no longer considered congested for these limits when the total number of buffers used by the queue drops below `buffer_queue_global_congestion_threshold[14:4]` minus `global_congestion_hysteresis[9:4]`.

Multicast/Broadcast Rate Limiting

The ET3028-50 has the ability to track the rate at which it is forwarding multicast and/or broadcast packets. If that rate exceeds a configurable maximum, the excess multicast and/or broadcast packets are discarded. A single rate-limiting function is shared by all Ethernet ports.

This packet rate limiting may be applied to either multicast packets or broadcast packets exclusively, or it may be applied to both multicast and broadcast packets. Multicast rate limiting is enabled by asserting the `multicast_rate_limit_en` bit in the `Multicast_Rate_Mode` register. Broadcast rate limiting is enabled by asserting the `broadcast_rate_limit_en` bit in the same register.

A leaky bucket style rate limiting function is used to measure the global rate of multicast packets and compare that rate against a configurable limit. If that limit is exceeded, a mask is applied to the destination bit map for all multicast and/or broadcast packets for as long as the condition persists.

The maximum multicast rate is established via the `Multicast_Rate_Decrement_Period` register. This register determines how often the `Multicast_Rate_Accumulator` is decremented. The period is calibrated in eight nanosecond units. So, setting this register to a value of one enables a multicast packet rate of 125,000,000 per second. A value of F_FFFF_{16} results in maximum rate of 119 multicast packets per second.

The `Multicast_Rate_Limit` register is used to establish the ET3028-50's multicast burst tolerance. A higher value in this register implies a higher tolerance for bursts of multicast packets. Once the value in `Multicast_Rate_Accumultor` exceeds that of `Multicast_Rate_Limit`, the multicast packet rate is determined to be in excess of the established limit.

When this condition is true, the mask in `Multicast_Rate_Discard_Mask` is applied during bridging operations to all received multicast packets. This mask is typically configured to exclude certain destinations from the masking operation. An example of such a destination is the supervisor queue that has been configured to be dedicated to the reception of BPDU packets.

Appendix B: Configuration (continued)

Other Networking Functions

This section describes the configuration requirements of a variety of additional packet handling methods supported by the ET3028-50.

IGMP Snooping

IGMP snooping enables the copying of IGMP packets to the supervisor for processing. These snooped packets are otherwise forwarded normally.

Each port may be individually enabled for IGMP snooping. This is accomplished by asserting the `igmp_snoop_en` bit in the `Layer_2_Mode` register that corresponds to the receive port of interest. Once a port has been enabled to have its receive traffic snooped for IGMP packets, all IGMP packets that are received have the port identified by the `Layer_2_Igmp_Snooping_Port` added to the packet's destination map. Typically, `Layer_2_Igmp_Snooping_Port` is configured to identify one of the supervisor's receive queues as the IGMP snooping port.

User-Port Snooping

User-port snooping makes it possible for the supervisor to receive copies of packets that are destined for a designated TCP or UDP destination port.

The TCP/UDP destination port number of interest is programmed into the `Layer_2_User_Port` register. Individual Ethernet receive ports are enabled for user-port snooping by asserting their corresponding `user_port_snoop_en` bits in the `Layer_2_Mode` register.

Packets received on the correctly enabled ports whose TCP or UDP destination port number matches the value in `Layer_2_User_Port` are copied to the port identified by the `Layer_2_User_Port_Snooping_Port` register in addition to being forwarded normally. Ordinarily, the user-port snooping port is chosen to be one of the supervisor's receive queues.

Cross-VLAN Routing

Communicating between VLANs typically requires the Layer 3 services of a router. The ET3028-50 includes the capability to detect that a packet has been disallowed from being forwarded due to VLAN restrictions.

Individual ports may be enabled to support the cross-VLAN routing feature. This is done by asserting the `supervisor_route_en` bits in the `Layer_2_Mode` register that correspond to the desired ports.

When a packet is received that ordinarily would have been forwarded to a transmit port if the destinations had not been inhibited by VLAN restrictions, the supervisor route port is added to the packet's destination map. This port is designated via the `Layer_2_Supervisor_Route_Port` register.

Packets forwarded to the supervisor route port are expected to be routed by the supervisor to their appropriate destinations.

Appendix B: Configuration (continued)

Other Networking Functions (continued)

Supervisor Packet Transmission

The ET3028-50 initiates *PCI* transfer cycles in order to read data structures stored within the supervisor's memory space that describe and contain packets to be transmitted onto one or more of the ET3028-50's Ethernet ports. The figure below shows the relationship between these various data structures.

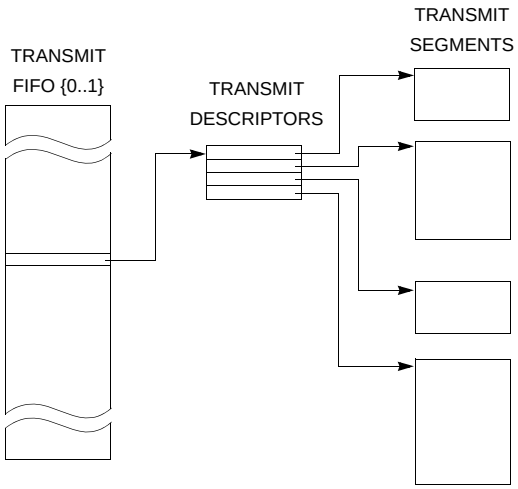


Figure 262. Supervisor Packet Transmission Data Structures

The following sections describe how to configure these various data structures and their related registers within the ET3028-50.

Transmit FIFOs. Two independent transmit FIFOs are supported. A strict priority relationship exists between the two with FIFO 1 being of higher priority than FIFO 0. Whenever FIFO 1 is nonempty, it preempts the servicing of FIFO 0.

These FIFOs contain lists of transmit descriptor pointers. One such pointer occupies each 32-bit word in a FIFO. These FIFOs reside in the supervisor's memory and are accessed automatically by the ET3028-50 via the *PCI* bus.

The physical boundaries of the FIFOs are defined by the `start_ptr[31:2]` and `end_ptr[31:2]` fields of the `Supervisor_Tx_Fifo_Limits` register. There are two instances of each of these fields; one for each FIFO.

The ET3028-50 starts processing a transmit FIFO at the location specified by the `start_ptr[31:2]` field and continues until reaching the location pointed to by the `last_ptr[31:2]` field. This field identifies the last valid entry in the respective transmit FIFO. The `last_ptr[31:2]` value is updated by the supervisor and interpreted by the ET3028-50. As the supervisor adds new entries to a transmit FIFO, it advances the value of `last_ptr[31:2]` to always point to the last valid entry in the FIFO. The advancement of this pointer causes the ET3028-50 to process the new FIFO contents.

The `end_ptr[31:2]` value defines the end of the physical block of memory containing the FIFO. Once the location indicated by `end_ptr[31:2]` has been used, operation of the FIFO continues at the location indicated by the value of `start_ptr[31:2]`.

Refer to the `Supervisor_Tx_Fifo_{0..1}` data structure description on page 48 for further information.

Appendix B: Configuration (continued)

Other Networking Functions (continued)

Transmit Descriptors. Each entry in the transmit FIFOs is a pointer to a transmit descriptor. Each transmit descriptor is a list of one or more 2-word (32-bit words) records that identify the various segments that make up a transmit packet.

Each 2-word structure consists of a pointer to a transmit segment, a segment length value, and a pair of flag bits: `last` and `indicate`.

The `last` flag is asserted if the current record is the last record of the packet descriptor. Once a record is encountered with its `last` flag asserted, packet descriptor processing is terminated and the last byte of the associate transmit segment is considered to be the last byte of the packet.

The `indicate` flag is asserted if it is desired to have an interrupt generated at the completion of packet transfer. This function is used to enable the supervisor to keep track of the transmission process. Interrupts enabled by this flag occur when the corresponding segment has completed its transfer across the *PCI* bus from the supervisor's memory to the ET3028-50's internal memory. This event may precede the actual transmission of the packet by a significant amount of time. In fact, the successful transfer of the packet (and its accompanying interrupt to the supervisor) is no assurance that the packet has been or ever will be transmitted. Certain events such as MAC flow control or maximum collisions on a half-duplex link may significantly delay or even prevent transmission.

Refer to the `Supervisor_Tx_Descriptor_{0..1}` data structure description on page 47 for further information.

Transmit Packet Segments. The transmit segments are simply contiguous bytes that are to be gathered up by the ET3028-50 as part of a transmit packet. The transmit packets must utilize a format that conforms to that shown for the `Supervisor_Tx_Packet` data structure. See page 49 for more details.

Refer to the `Supervisor_Tx_Packet_Segment_{0..1}` data structure description on page 50 for further information.

Supervisor Packet Reception

Eight of the 232 queues within the ET3028-50 are dedicated to the supervisor for packet reception purposes. These eight queues correspond to bits 50 through 57 of the 38-bit destination map that is computed as part of the bridging process.

There is a one-to-one correlation between these eight internal queues and the eight receive FIFOs that are maintained by the ET3028-50 in the supervisor's memory space across the *PCI* bus.

Appendix B: Configuration (continued)

Other Networking Functions (continued)

Defining the FIFOs. The `Supervisor_Rx_Fifo_Limits` register is used to establish the physical extent of the eight packet receive FIFOs in the supervisor memory space. The `rx_fifo_start_ptr[31:2]` and `rx_fifo_end_ptr[31:2]` fields point to the first and last 32-bit words of each receive FIFO. The ET3028-50 deposits receive packets into contiguous locations between these two limits.

The `rx_fifo_first_ptr[31:2]` field is used by the supervisor to protect receive packets that have been transferred across the *PCI* bus into the supervisor's memory and which the supervisor has not yet finished using. The supervisor sets `rx_fifo_first_ptr[31:2]` to point to the first word of the packet that it is currently working with. The ET3028-50 will not advance its write pointer beyond the point identified by `rx_fifo_first_ptr[31:2]`, thus preventing the overrunning of FIFO data.

Managing the FIFOs. As packets are transferred into a supervisor's receive FIFO, two pointers are advanced by the ET3028-50 to inform the supervisor of the state of the receive transfer process. These pointers reside in the `Supervisor_Rx_Fifo_Ptr` register. There is one set of pointers for each of the eight queues.

The `rx_fifo_wr_ptr[31:2]` value indicates the location of supervisor memory currently being addressed during write operations. This pointer is for informational purposes only and need not be monitored during normal operation.

The `rx_fifo_last_ptr[31:2]` value points to the start of the newest complete packet in the corresponding receive FIFO. Once the supervisor has processed the packet identified by this pointer, the FIFO is considered empty.

Statistics

The extent to which statistics require any configuration is that they must be reset to zero or some other starting value prior to normal operation of the device. When any of the ET3028-50's statistics counters reach their maximum value, they roll over to zero without notification and continue counting. Consequently, the statistics counters must be sampled often enough such that the sample period is always less than the counters' minimum roll over period.

Appendix B: Configuration (continued)

Other Device Functions

Device Reset

If possible, the ET3028-50 should be supplied with a reset signal (RST_N) that is separate from the powerup reset for the rest of the system so that the ET3028-50 may be independently reset. If the device is powered up and the core clock (REFCLK_CORE) is stable, a low going pulse of 200 ns or greater is sufficient to reset the device. During powerup, the reset signal should be held low for at least 200 ns after power has ramped and REFCLK_CORE is stable. Figure 263. illustrates the desired reset operation. In this figure, RST_N is represented as both a continuously low signal and a low-going pulse. The continuously low signal represents the reset during powerup. The pulse is representative of a reset when the device is powered up and REFCLK_CORE is stable.

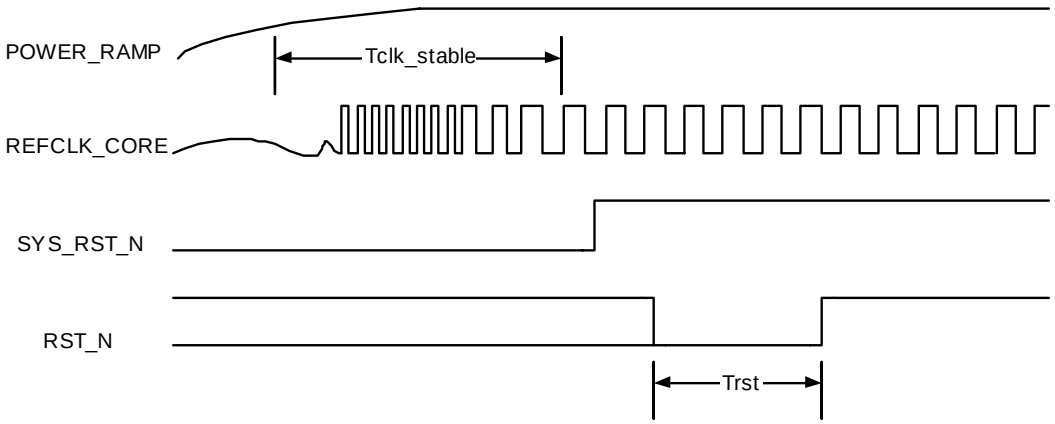


Figure 263. RST_N Operation

Table 342. RST_N Timing

Parameter	Description	Value
Tclk_stable	Period needed for core clock stabilization	System-dependent
Trst	Period needed after power ramps and core clock is stable	≥200 ns

Appendix B: Configuration (continued)

Other Device Functions (continued)

PCI Initialization

For the PCI configuration space, the first 16 bytes are defined the same for all device types. In the ET3028-50, the Header Type (offset 0x0E) is hardcoded to 0x00 and indicates that the configuration space header is type 0. This type 0 header is represented in Figure 264.

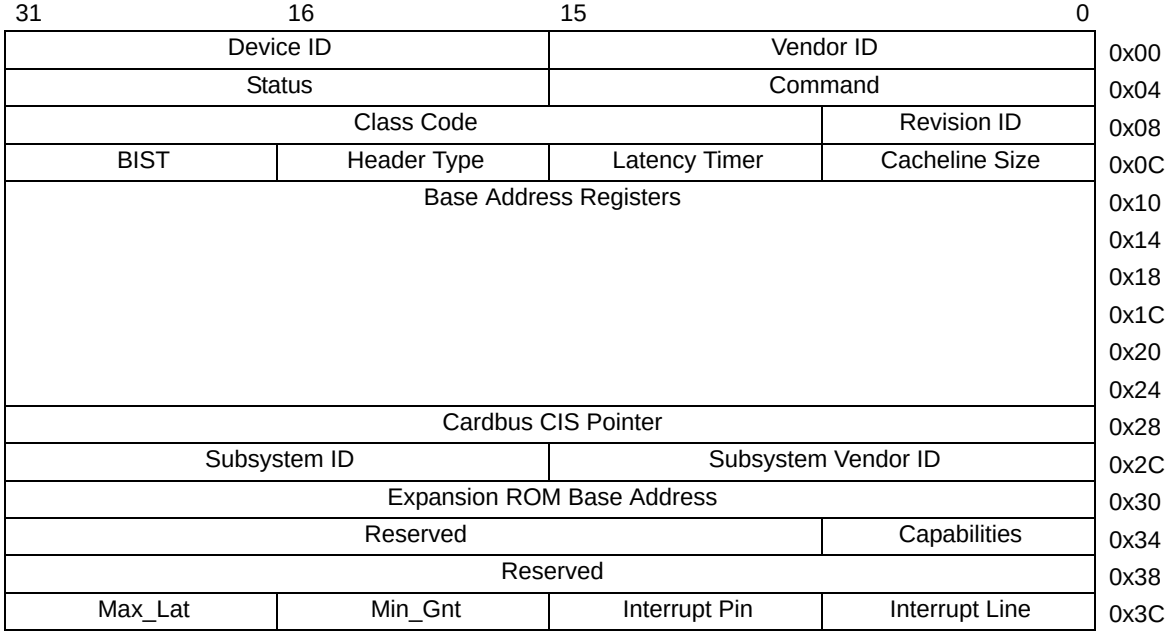


Figure 264. Type 0 Configuration Space Header

The PCI specification requires implementation of five read-only fields. These include device ID, vendor ID, revision ID, header type, and class code. Class code consists of three fields: base class code, subclass code, and programming interface code. As mentioned above, the value of the header type field is 0x00. The values of these remaining fields are as follows:

- Device ID (offset 0x02) = 0x1107
- Vendor ID (offset 0x00) = 0x0700
- Revision ID (offset 0x08) = 0x01
- Class code
  — Base class code (offset 0x09) = 0x06
  — Subclass code (offset 0x0A) = 0x00
  — Programming interface code (offset 0x0B) = 0x00

For initialization of the configuration space, program the command field (offset 0x04) to 0x0006. In addition, program the base address register (offset 0x10) to support a 1 Mbyte memory space using a value of 0xXXX00000, where **X** represents any arbitrary hexadecimal value. The least-significant nibble of the base address field has a read-only value of 0xC, which specifies that the base address is prefetchable and maps into memory space.

Initially, per the specification, the PCI bus is little-endian; however, the ET3028-50 allows the reprogramming of the bus to big-endian after PCI initialization. See the Supervisor Endian section below for information on reprogramming the endianness. For more information on the PCI bus, see the PCI Local Bus Specification Rev. 3.0, February 3, 2004.

Appendix B: Configuration (continued)

Other Device Functions (continued)

Supervisor Endian

The supervisor's *PCI* bus can operate in either a big-endian or a little-endian fashion. This operating mode is set via the `big_endian` bit in the `Supervisor_Endian` register. Clearing this bit to zero places the *PCI* bus in its normal operating mode: little endian. Setting the bit causes the *PCI* bus to operate in big-endian manner. It is recommended that 0000_0000_{16} or $FFFF\_FFFF_{16}$ be written to this register and to not bother with 8000_0000_{16} or 0000_0001_{16}. Doing so guarantees that the `big_endian` bit is set as desired.

Interrupts

On-chip events may result in indications to the supervisor. These indications may, in turn, result in a hardware interrupt. This hierarchy of events, indications and interrupts is presented via a series of registers, records, and fields that control which events may become indications and, in turn, which indications may become interrupts.

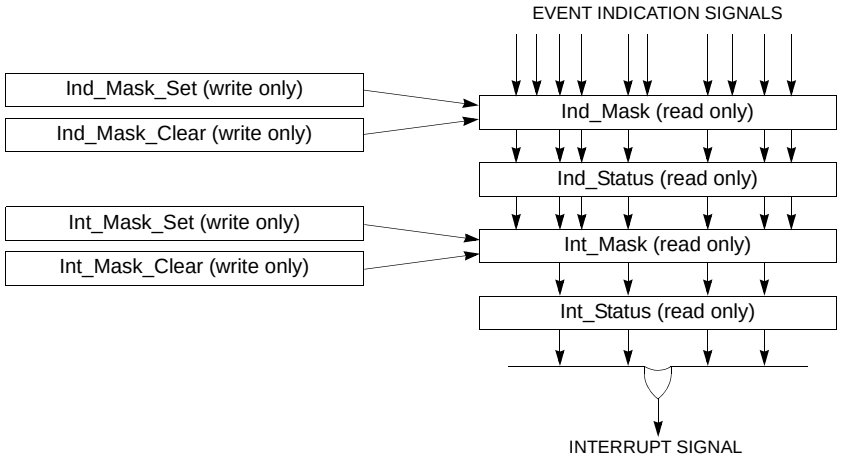


Figure 265. Indications and Interrupts

Signals that indicate the occurrence of the various events must pass through a mask before being presented to an indication status register that is readable by the supervisor. This mask uses two write-only registers to set and clear the mask bits. This technique of manipulating mask bits is done to make it possible to turn on or off a single bit or a subset of bits with an atomic operation that cannot be interrupted.

The unmasked indication bits are then passed through an interrupt mask. This mask is managed in much the same manner as the indication masks. Those indications that pass through the interrupt masks are presented via the interrupt status registers to the supervisor. Bits asserted in the interrupt status register are also ORed together to form a hardware interrupt to the supervisor.

Packet Buffer Scrubbing

To guard against the possibility of a buffer within the packet buffer becoming lost from the free list and, hence, no longer available for packet storage operations, the ET3028-50 includes a packet scrubbing mechanism. This mechanism is activated by writing the buffer number of a presumed-missing buffer into the `missing_buffer[14:0]` field of the `Packet_Buffer_Scrub` register. The ET3028-50 takes that buffer number and determines if it exists on the free list or any of the transmit queues or descriptors. If the buffer cannot be found, it is added to the free list as a new entry. During the searching operation, the `scrubbing` bit is asserted by the ET3028-50 in the `Packet_Buffer_Scrub` register.

Ordering Information

Table 343. Ordering Information

Device	Description	Part Number	Comcode
ET3028-50	Single-Chip 28 x 1 Gbit/s Layer 2 Ethernet Switch, Revision B	ET3028-50B-DB	700079717
ET3028-50	Single-Chip 28 x 1 Gbit/s Layer 2 Ethernet Switch, Revision B1	ET3028-50B1-DB	711002814
ET3028-50	Single-Chip 28 x 1 Gbit/s Layer 2 Ethernet Switch, Revision C	ET3028-50C-DB	711006779
ET3028-50	Lead-Free Single Chip 28 x 1 Gbit/s Layer 2 Ethernet Switch, Revision B	L-ET3028-50B-DB	700079859
ET3028-50	Lead-Free Single Chip 28 x 1 Gbit/s Layer 2 Ethernet Switch, Revision C	L-ET3028-50C-DB	711006780

Related Documentation

Table 344. Related Documentation

Device	Description	Document Type
ET1011	Gigabit Ethernet Transceiver	Product Brief Data Sheet Application Note
ET1310	Gigabit Ethernet Controller	
ET1081	Gigabit Ethernet Octal PHY	Product Brief Data Sheet
ET4028-50	Single-Chip 28 x 1 Gbit/s Layer 2+ Ethernet Switch	Product Brief Data Sheet Application Note
ET4048-50	Single-Chip 48 x 1 Gbit/s Layer 2+ Ethernet Switch	
ET4128-50	Single-Chip 28 x 1 Gbit/s + 2x 10 Gbits/s Layer 2+ Ethernet Switch	
ET4148-50	Single-Chip 48 x 1 Gbit/s + 2x 10 Gbits/s Layer 2+ Ethernet Switch	
ET3048-50	Single-Chip 48 x 1 Gbit/s Layer 2 Ethernet Switch	

PCI is a trademark of PCI-SIG Corporation.
IEEE and *802* are registered trademarks of the Institute of Electrical and Electronics Engineers, Inc.

For additional information, contact your Agere Systems Account Manager or the following:

INTERNET:	Home: **http://www.agere.com** Sales: **http://www.agere.com/sales**
E-MAIL:	**docmaster@agere.com**
N. AMERICA:	Agere Systems Inc., Lehigh Valley Central Campus, Room 10A-301C, 1110 American Parkway NE, Allentown, PA 18109-9138 **1-800-372-2447**, FAX 610-712-4106 (In CANADA: **1-800-553-2448**, FAX 610-712-4106)
ASIA:	CHINA: **(86) 21-54614688** (Shanghai), **(86) 755-25881122** (Shenzhen), **(86) 10-65391096** (Beijing) JAPAN: **(81) 3-5421-1600** (Tokyo), KOREA: **(82) 2-767-1850** (Seoul), SINGAPORE: **(65) 6741-9855**, TAIWAN: **(886) 2-2725-5858** (Taipei)
EUROPE:	**Tel. (44) 1344 865 900**

April 2006
DS06-098GSWC (Replaces DS05-176GSWC)