



Link Street®

88E6321/88E6320 Functional Specification

Doc. No. MV-S109503-00, Rev. --

August 6, 2013

CONFIDENTIAL

Document Classification: Proprietary Information



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Document Conventions

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

Document Status

Doc Status: Draft	Technical Publication: 0.03
-------------------	-----------------------------

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

Disclaimer

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Marvell. Marvell retains the right to make changes to this document at any time, without notice. Marvell makes no warranty of any kind, expressed or implied, with regard to any information contained in this document, including, but not limited to, the implied warranties of merchantability or fitness for any particular purpose. Further, Marvell does not warrant the accuracy or completeness of the information, text, graphics, or other items contained within this document.

Marvell products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use Marvell products in these types of equipment or applications.

With respect to the products described herein, the user or recipient, in the absence of appropriate U.S. government authorization, agrees:

- 1) Not to re-export or release any such information consisting of technology, software or source code controlled for national security reasons by the U.S. Export Control Regulations ("EAR"), to a national of EAR Country Groups D:1 or E:2;2) Not to export the direct product of such technology or such software, to EAR Country Groups D:1 or E:2, if such technology or software and direct products thereof are controlled for national security reasons by the EAR; and,
- 3) In the case of technology controlled for national security reasons under the EAR where the direct product of the technology is a complete plant or component of a plant, not to export to EAR Country Groups D:1 or E:2 the direct product of the plant or major component thereof, if such direct product is controlled for national security reasons by the EAR, or is subject to controls under the U.S. Munitions List ("USML").

At all times hereunder, the recipient of any such information agrees that they shall be deemed to have manually signed this document in connection with their receipt of any such information.

Copyright © 1999–2013, Marvell International Ltd. All rights reserved. Alaska, ARMADA, CarrierSpan, Kinoma, Link Street, LinkCrypt, Marvell logo, Marvell, Moving Forward Faster, PISC, Prestera, Qdeo (for chips), QDEO logo (for chips), QuietVideo, Virtual Cable Tester, Xelerated, and Yukon are registered trademarks of Marvell or its affiliates. Avanta, Avastar, DragonFly, HyperDuo, Kirkwood, Marvell Smart, Qdeo, QDEO logo, The World as YOU See It, Vmeta and Wirespeed by Design are trademarks of Marvell or its affiliates.

Patent(s) Pending—Products identified in this document may be covered by one or more Marvell patents and/or patent applications.

Preface

About this Document

The Link Street® 88E6321/88E6320 Functional Specification provides a description of the switch core and PHYs, and includes the related register tables.

Related Documents:

- Link Street® 88E6321/88E6320 Datasheet, which provides a features list and overview describing the features in the 88E6321 and 88E6320. It also provides the pin description, pin map, and electrical specifications (Document Number: MV-S108695-00)



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 1 shows the 88E6321/88E6320 device feature differences.

Table 1: 88E6321/88E6320 Functional Specification Device Differences

		88E6321	88E6320
Features	# GE PHYs	2	2
	# SERDES/GMII	2	2
	# RGMII/MII/RMII	3	3
	# GMII (shared between Port 2 and Port 6)	1	1
	802.3az (EEE) Support	Yes	Yes
	Packet Buffer Memory	1 Mbit	1 Mbit
	Jumbo Frame Support	10K Bytes	10K Bytes
	# MAC Addresses	8K	8K
	Queues per Port	4	4
	802.1p, Port, TOS/DS, IPv6, TC, MAC	Yes	Yes
QoS	Programmable Weighting	Yes	Yes
	Port-based VLANs	Yes	Yes
	802.1Q VLANs	4096	4096
	Double Tagging (Q in Q)	Yes	Yes
	Wake On LAN/Wake On Frame	Yes	Yes
	Remote Mgmt/Ethertype DSA	Yes	Yes
	Layer 2 PCLs	Yes	Yes
	802.1D/s/w Spanning Tree	Yes	Yes
	802.1X Port & MAC Authentication	Yes	Yes
	Port Mirroring/Port Trunking	Yes	Yes
VLAN	IGMP/MLD Snooping	Yes	Yes
	802.1AS/Qat/Qav	Yes	Yes
	Timing App I/F (TAI)	Yes	Yes
	IEEE 1588v2	Yes	Yes
	Ingress Rate Limiting Resources	5/port	5/port
	Egress Rate Shaping	Yes	Yes
	GPIO Pins (shared)	15	15
	Integrated Voltage Regulators	Yes	Yes
	Synchronous Ethernet	Yes	No
	256 Entry TCAM	Yes	No
Management	Cut Through Switching	Yes	No
	Typical Power	0.9W	0.9W
	Package	108 QFN	108 QFN

Table of Contents

1	Switch Core Functional Description.....	23
2	Basic Switch Functions	24
2.1	Physical Switch Data Flow	24
2.2	Physical Interface	25
2.3	Media Access Controllers (MAC)	25
2.3.1	Backoff.....	26
2.3.2	Half-duplex Flow Control	26
2.3.3	Full-duplex Flow Control	26
2.3.4	Forcing Flow Control in the MAC	27
2.3.5	Jamming Control - Egress Limit.....	28
2.3.6	Jamming Control - Ingress Limit	28
2.3.7	Forcing Link, Speed, and Duplex in the MAC	28
2.3.8	MAC Based RMON/Statistics Counters.....	29
2.3.9	Policy Based RMON/Statistics Counters	33
2.4	Basic Switch Operation	34
2.4.1	Lookup Engine.....	34
2.4.2	Address Searching or Translation	35
2.4.3	Automatic Address Learning.....	35
2.4.4	Hardware Address Learn Limiting	36
2.4.5	Automatic Address Aging	37
2.4.6	CPU Directed Address Learning and Purging	37
2.4.7	802.1X Source MAC Address Checking	38
2.4.8	Multiple Address Database Support (FID)	39
3	Normal Network Ports	40
3.1	Ingress Policy	40
3.1.1	Port States Filtering for 802.1D Spanning Tree	40
3.1.2	Source Address Filtering	41
3.1.3	Layer 2 Policy Control Lists	44
3.2	VLANS	47
3.2.1	Port Based VLANs	47
3.2.2	802.1Q VLANs	50
3.2.3	802.1s Per VLAN Spanning Tree	53
3.3	Special Frame Handling	55
3.3.1	Switching Frames Back to their Source Port	55
3.3.2	Tunneling Frames through VLANs	55
3.3.3	ARP Mirroring	55
3.3.4	IGMP Trapping or Snooping	56



Link Street®

88E6321/88E6320 Functional Specification

3.4	Quality of Service (QoS) Classification	58
3.4.1	IEEE Tagged Frame Priority Extraction.....	58
3.4.2	IPv4 and IPv6 Frame Priority Extraction.....	59
3.4.3	Default Priority	60
3.4.4	Initial Priority Selection	60
3.4.5	Frame Type Priority Override	62
3.4.6	Layer 2 Priority Override	63
3.5	Port Based Ingress Rate Limiting (PIRL)	64
3.6	Queue Controller	69
3.6.1	Frame Latencies	69
3.6.2	No Head-of-Line Blocking	69
3.6.3	QoS with and without Flow Control.....	69
3.6.4	Guaranteed Frame Delivery without Flow Control.....	70
3.6.5	The Queues	70
3.6.6	Per Port Fixed or Weighted Priority	71
3.6.7	Programmable Weighting Table	72
3.7	Embedded Memory	73
3.8	Egress Policy	74
3.8.1	Forward Unknown/Secure Port.....	74
3.8.2	Forward Unknown for Multicasts	74
3.8.3	Secure 802.1Q VLANs	75
3.8.4	Tagging and Untagging Frames	75
3.8.5	Egress Rate Shaping	78
4	Provider Mode Ports	81
4.1	Customer to Provider	82
4.2	Provider to Customer	83
4.2.1	Provider VID Processing.....	83
4.2.2	Provider QoS Processing	83
4.3	Customer to Customer	85
4.4	Provider to Provider	85
4.5	Recursive Provider Tag Stripping.....	86
4.6	Restrictions on Provider Ports.....	86
4.7	Restrictions on Customer Ports.....	86
5	Distributed Switch Architecture (DSA) Ports.....	87
5.1	Forward DSA Tag	87
5.2	To_CPU DSA Tag	90
5.3	From_CPU DSA Tags	92
5.4	To_Sniffer DSA Tag	94
5.5	Cross-chip Features Using DSA Links	96
5.5.1	Cross-chip Flow Control	96
5.5.2	Cross-chip 802.1Q VLANs	96
5.5.3	Cross-chip Port Based VLANs	97
5.6	Switch Handling of DSA MGMT Frames	99

5.7	Proper Usage of DSA Tag Ports	100
5.8	Secure Control Technology (SCT).....	101
5.9	Ether Type DSA Tag	102
6	Advanced Switch Functions.....	103
6.1	Management Frames to and from the CPU	104
6.2	Spanning Tree Support	105
6.3	Ingress MGMT/BPDU Frame Detection	106
6.3.1	Reserved Multicast Address Support	106
6.3.2	New and Proprietary Protocol Support	106
6.4	Other Ingress MGMT Frame Detection	107
6.5	MGMT Frames to Normal or Provider Egress.....	107
6.6	Proper Connection to a Management CPU.....	108
6.7	Proper Connection to a Router	108
6.7.1	Switch Ingress Header for Routers	108
6.7.2	Switch Egress Header	110
6.8	MUX'ing or Ignoring Address Translation.....	112
6.8.1	Passing Frames to a Router	112
6.8.2	Operational, Administration, and Maintenance (OAM) Loopback	112
6.9	Port Monitoring Support	113
6.10	Port Trunking Support	115
6.10.1	Trunk Address Learning	115
6.10.2	Trunk Address Searching	115
6.10.3	Trunk Mapping	115
6.10.4	Load Balancing	115
6.11	Precise Time Protocol (PTP).....	117
6.12	Interrupt Controller	120
6.12.1	Device Interrupts.....	120
7	Accessing Data Structures	121
7.1	Ternary Content Addressable Memory (TCAM) - (88E6321 Only)	121
7.1.1	Format of the TCAM Database	122
7.1.2	Reading the TCAM Database	125
7.1.3	Loading an Entry in the TCAM Database	127
7.1.4	Moving an Entry	129
7.1.5	Updating an Entry	129
7.1.6	Flushing Entries	129

7.2	VLAN Translation Unit Operations	130
	7.2.1 Format of the VTU Database	131
	7.2.2 Reading the VLAN Database.....	133
	7.2.3 Loading and Purging an Entry in the VLAN Database.....	134
	7.2.4 Flushing Entries	134
	7.2.5 Servicing VTU Violations	135
	7.2.6 Format of the STU Database.....	136
	7.2.7 Reading the SID Database	138
	7.2.8 Loading and Purging an Entry in the STU Database	139
	7.2.9 Flushing Entries	140
7.3	Address Translation Unit Operations	141
	7.3.1 Format of the ATU Database	142
	7.3.2 Reading the Address Database.....	144
	7.3.3 Loading & Purging an Entry in the Address Database	145
	7.3.4 Flushing Entries	146
	7.3.5 Moving or Removing Single Port Mappings.....	146
	7.3.6 Servicing ATU Violations	147
	7.3.7 ATU Statistics	148
8	Remote Management	150
8.1	Request for Frame Format - Layer 2 and DSA Portion	151
	8.1.1 RMU and Ether type DSA.....	152
	8.1.2 RMU and Marvell® Header.....	152
8.2	Response Frame Format - Layer 2 and DSA Portion	153
	8.2.1 Restrictions of Remote Management	153
8.3	Request Frame Format - Layer 3	155
	8.3.1 The Initial Request Frame - GetID	155
8.4	Response Frame Format - Layer 3	156
	8.4.1 The Initial Response Frame - GotID	156
	8.4.2 Error Handling.....	156
8.5	Supported Requests and Responses	157
	8.5.1 GetID (non-destructive)	157
	8.5.2 Dump ATU (non-destructive)	158
	8.5.3 Dump MIBs (non-destructive)	159
	8.5.4 Dump MIBs and Clear (destructive).....	160
	8.5.5 Read/Write Register (may be destructive).....	160
	8.5.6 Error Response Frame (non-destructive)	163

9	PHY and SERDES Functional Specifications	164
9.1	PHY Functional Description	164
9.1.1	Media Interface	164
9.1.2	Loopback	167
9.1.3	Synchronizing FIFO	170
9.1.4	Power Management.....	171
9.1.5	SERDES Manual Power Down.....	173
9.1.6	Auto-Negotiation	174
9.1.7	Downshift Feature.....	178
9.1.8	Fast 1000BASE-T Link Down Indication.....	178
9.1.9	Advanced Virtual Cable Tester® (VCT™).....	179
9.1.10	Data Terminal Equipment (DTE) Detect.....	183
9.1.11	Energy Efficient Ethernet (EEE)	184
9.1.12	CRC Error Counter and Packet Counter	184
9.1.13	Packet Generator.....	185
9.1.14	MDI/MDIX Crossover	186
9.1.15	Polarity Correction	187
9.1.16	FLP Exchange Complete with No Link	187
9.1.17	Interrupt	188
9.1.18	Temperature Sensor.....	189
9.2	SERDES Functional Description	190
9.3	1.25 GHz SERDES Interface	190
9.3.1	Electrical Interface	190
9.3.2	SGMII Speed and Link.....	191
9.3.3	SGMII TRR Blocking.....	191
9.3.4	False SERDES Link Up Prevention	191
9.4	Loopback.....	192
9.4.1	Synchronous SERDES Loopback	192
9.5	Resets	193
9.6	Auto-Negotiation	194
9.6.1	1000BASE-X Auto-Negotiation	194
9.6.2	SGMII Auto-Negotiation	194
9.7	1.25G PRBS Generator and Checker	196
9.8	Unidirectional Transmit	196
9.9	Synchronous Clocking Support (88E6321 Device Only).....	197
9.9.1	Recovered Clock	197
9.9.2	Clock Select.....	197
9.10	MDC/MDIO.....	198
9.10.1	Clause 22 MDC/MDIO Management Interface	198
9.10.2	Extended Register Access.....	199
9.10.3	Clause 45 MDC/MDIO Management Interface (XMDIO).....	200
9.10.4	Clause 22 Access to Clause 45 MDIO Manageable Device (MMD).....	200
9.10.5	Clause 45 Access to Clause 22 Registers	202
9.10.6	Preamble Suppression	202
10	Switch Register Description	203
10.1	Register Types	210
10.2	Multi-chip Addressing Mode	211
10.3	Single-chip Addressing Mode	213



10.4	Switch Port Registers	215
10.4.1	Switch Global 1 Registers	266
10.4.2	Switch Global 2 Registers	293
10.5	Port Ingress Rate Limiting (PIRL) Registers	365
10.6	AVB Registers	374
10.6.1	Precise Timing Protocol (PTP) Registers	375
10.6.2	AVB Registers	409
10.6.3	Qav Registers	417
10.6.4	Switch Global 3 Registers - TCAM (88E6321) Only	425
10.6.5	Common Global 3 Registers for all TCAM Pages:.....	426
10.6.6	Global 3 Registers for TCAM Page 0:.....	428
10.6.7	Global 3 Registers for TCAM Page 1	434
10.6.8	Global 3 Registers for TCAM Page 2:.....	436
11	PHY and SERDES Register Description	443
11.1	PHY Register Description	443
11.1.1	MDIO Register Description	444
11.1.2	XMDIO Register Description.....	488
11.2	SERDES Register Description	492
12	EEPROM Programming Format	512

List of Tables

Table 1:	88E6321/88E6320 Functional Specification Device Differences	4
Table 2:	Pause Frame Format.....	27
Table 3:	Ingress Statistics Counters.....	30
Table 4:	Egress Statistics Counters.....	32
Table 5:	Port State Options	41
Table 6:	VLANTable Settings for Figure 6	48
Table 7:	VLANTable Settings for Figure 7	49
Table 8:	Example VID Assignment Summary.....	51
Table 9:	802.1s Port State Options.....	53
Table 10:	Initial QPri and FPri Selection	61
Table 11:	Forward DSA Tag Fields	88
Table 12:	Src_Tagged vs. Normal Network Egress Mode Actions	89
Table 13:	To_CPU DSA Tag Fields.....	91
Table 14:	To_CPU Code Meanings.....	91
Table 15:	From_CPU DSA Tag Fields.....	93
Table 16:	From_CPU DSA Tag Fields.....	95
Table 17:	Secure Control Technology Example	101
Table 18:	Ether Type DSA Tag Fields	102
Table 19:	Egress Header Fields - Header Type = 00 (Original Header).....	110
Table 20:	Egress Header Fields - Header Type = 01 (Single chip MGMT Header)	111
Table 21:	TCAM Operation Register	121
Table 22:	TCAM Entry Format - TCAM Key and Frame Data with Masks.....	123
Table 23:	TCAM Entry Format - Action Data	124
Table 24:	TCAM Get Next Operation Register Usage	125
Table 25:	TCAM Read Operation Register Usage	126
Table 26:	TCAM Load Operation Register Usage	128
Table 27:	VTU Operation Register	130
Table 28:	VTU Entry Format.....	131
Table 29:	VTU Get Next Operation Register Usage	133
Table 30:	VTU Load/Purge Operation Register Usage	134
Table 31:	VTU Get/Clear Violation Register Usage	135
Table 32:	STU Entry Format.....	137
Table 33:	STU Get Next Operation Register Usage.....	138
Table 34:	STU Load/Purge Operation Register Usage	139
Table 35:	Egress Header Fields	141
Table 36:	ATU Data Bits	142
Table 37:	ATU Get Next Operation Register Usage	144
Table 38:	ATU Load/Purge Operation Register Usage	145
Table 39:	ATU Get/Clear Operation Register Usage	148



Link Street®

88E6321/88E6320 Functional Specification

Table 40:	MAC Speed Control	168
Table 41:	FIFO Status Bits	170
Table 42:	Power State after Exiting Power Down.....	172
Table 43:	SERDES Power Down Control Bits	173
Table 44:	Reverse Auto-Negotiation Priority Resolution	175
Table 45:	Reverse Auto-Negotiation, Disable 1000BASE-T, and Downshift Behavior.....	176
Table 46:	Reverse Auto-Negotiation Registers	177
Table 47:	Registers for DTE Power	183
Table 48:	Media Dependent Interface Pin Mapping	186
Table 49:	Misc Test Register (Temperature Sensor Register 1)	189
Table 50:	Temperature Sensor Register (Temperature Sensor Register 2).....	189
Table 51:	Fiber Noise Filtering.....	191
Table 52:	Reset Control Bits	193
Table 53:	SGMII Auto-Negotiation modes	195
Table 54:	1.25 GHz SERDES PRBS Registers	196
Table 55:	Serial Management Interface Protocol	198
Table 56:	Page Address	199
Table 57:	Extensions for Management Frame Format for Indirect Access.....	200
Table 58:	MMD Access Control Register.....	200
Table 59:	MMD Access Address/Data Register	201
Table 60:	Clause 45 Access to Clause 22 Registers Example.....	202
Table 61:	Register Map—Multi-Chip Addressing Mode.....	203
Table 62:	Register Map	203
Table 63:	SMI Command Register	212
Table 64:	SMI Data Register	212
Table 65:	Port Status Register.....	216
Table 66:	Interface Configuration Matrix.....	219
Table 67:	Physical Control Register	221
Table 68:	Jamming Control Register	224
Table 69:	Switch Identifier Register	225
Table 70:	Port Control Register	226
Table 71:	Port Control 1.....	232
Table 72:	Port Based VLAN Map.....	232
Table 73:	Default Port VLAN ID & Priority	234
Table 74:	Port Control 2 Register	235
Table 75:	Egress Rate Control	239
Table 76:	Egress Rate Control 2	240
Table 77:	Port Association Vector	242
Table 78:	Port ATU Control	244
Table 79:	Priority Override Register	246
Table 80:	Policy Control Register	251
Table 81:	Port E Type.....	254
Table 82:	LED Control	255
Table 83:	LED Port to Physical Mapping	255

Table 84:	LED 0 & 1 Control, Register Index: 0x00 of LED Control for Ports 0 to 4	256
Table 85:	LED 0 & 1 Control, Register Index: 0x00 of LED Control for Alternate LEDs Port 5	257
Table 86:	Stretch and Blink Rate Control, Register Index: 0x06 of LED Control	259
Table 87:	Port 3 Special Control, Register Index: 0x07 of LED Control on Port 3.....	260
Table 88:	Port 4 Special Control, Register Index: 0x07 of LED Control on Port 4.....	260
Table 89:	Port 5 Special Control, Register Index: 0x07 of LED Control on Port 5.....	261
Table 90:	Port IEEE Priority Remapping Registers	262
Table 91:	Port IEEE Priority Remapping Registers	262
Table 92:	Queue Counter Registers	263
Table 93:	Debug Counter	264
Table 94:	Cut Through Register	265
Table 95:	Switch Global Status Register	267
Table 96:	ATU FID Register	268
Table 97:	VTU FID Register	269
Table 98:	VTU SID Register	269
Table 99:	Switch Global Control Register	270
Table 100:	VTU Operation Register	272
Table 101:	VTU VID Register	273
Table 102:	VTU/STU Data Register Ports 0 to 3 for VTU Operations	273
Table 103:	VTU/STU Data Register Ports 0 to 3 for STU Operations	274
Table 104:	VTU/STU Data Register Ports 4 to 5 for VTU Operations	275
Table 105:	VTU/STU Data Register Ports 4 to 5 for STU Operations	276
Table 106:	VTU/STU Data Register for VTU Operations.....	276
Table 107:	ATU Control Register.....	277
Table 108:	ATU Operation Register	278
Table 109:	ATU Data Register.....	280
Table 110:	ATU MAC Address Register Bytes 0 & 1	281
Table 111:	ATU MAC Address Register Bytes 2 & 3	281
Table 112:	ATU MAC Address Register Bytes 4 & 5.....	281
Table 113:	IEEE-PRI Register	282
Table 114:	IP Mapping Table.....	283
Table 115:	Monitor Control	284
Table 116:	Total Free Counter.....	286
Table 117:	Global Control 2	287
Table 118:	Stats Operation Register	289
Table 119:	Stats Counter Register Bytes 3 & 2	292
Table 120:	Stats Counter Register Bytes 1 & 0	292
Table 121:	Interrupt Source Register.....	294
Table 122:	Interrupt Mask Register	295
Table 123:	MGMT Enable Register 2x	296
Table 124:	MGMT Enable Register 0x	296
Table 125:	Flow Control Delay Register	297
Table 126:	Switch Management Register	298
Table 127:	Device Mapping Table Register.....	300



Link Street®

88E6321/88E6320 Functional Specification

Table 128: Trunk Mask Table Register	301
Table 129: Trunk Mapping Table Register	301
Table 130: Ingress Rate Command Register	302
Table 131: Ingress Rate Data Register	303
Table 132: Cross-chip Port VLAN Register	304
Table 133: Cross-chip Port VLAN Data Register	305
Table 134: Switch MAC/WoL/WoF Register	306
Table 135: Switch MAC Byte 0, Register Index: 0x00 of Switch MAC/WoL/WoF	307
Table 136: Switch MAC Byte 1, Register Index: 0x01 of Switch MAC/WoL/WoF	307
Table 137: Switch MAC Byte 2, Register Index: 0x02 of Switch MAC/WoL/WoF	307
Table 138: Switch MAC Byte 3, Register Index: 0x03 of Switch MAC/WoL/WoF	307
Table 139: Switch MAC Byte 4, Register Index: 0x04 of Switch MAC/WoL/WoF	307
Table 140: Switch MAC Byte 5, Register Index: 0x05 of Switch MAC/WoL/WoF	308
Table 141: Wake on Frame Int Source, Register Index: 0x0C of Switch MAC/WoL/WoF	308
Table 142: Wake on Frame Int Enable, Register Index: 0x0E of Switch MAC/WoL/WoF	309
Table 143: WoL MAC Byte 0, Register Index: 0x10 of Switch MAC/WoL/WoF	309
Table 144: WoL MAC Byte 1, Register Index: 0x11 of Switch MAC/WoL/WoF	309
Table 145: WoL MAC Byte 2, Register Index: 0x12 of Switch MAC/WoL/WoF	310
Table 146: WoL MAC Byte 3, Register Index: 0x13 of Switch MAC/WoL/WoF	310
Table 147: WoL MAC Byte 4, Register Index: 0x14 of Switch MAC/WoL/WoF	310
Table 148: WoL MAC Byte 5, Register Index: 0x15 of Switch MAC/WoL/WoF	310
Table 149: WoL Password Byte 0, Register Index: 0x16 of Switch MAC/WoL/WoF	310
Table 150: WoL Password Byte 1, Register Index: 0x17 of Switch MAC/WoL/WoF	310
Table 151: WoL Password Byte 2, Register Index: 0x18 of Switch MAC/WoL/WoF	311
Table 152: WoL Password Byte 3, Register Index: 0x19 of Switch MAC/WoL/WoF	311
Table 153: WoL Password Byte 4, Register Index: 0x1A of Switch MAC/WoL/WoF	311
Table 154: WoL Password Byte 5, Register Index: 0x1B of Switch MAC/WoL/WoF	311
Table 155: WoL Control Byte 0, Register Index: 0x1C of Switch MAC/WoL/WoF	312
Table 156: WoL Control Byte 1, Register Index: 0x1D of Switch MAC/WoL/WoF	312
Table 157: WoL Control Byte 2, Register Index: 0x1E of Switch MAC/WoL/WoF	312
Table 158: Wake Interrupt Source, Register Index: 0x1F of Switch MAC/WoL/WoF	313
Table 159: ATU Stats Register	314
Table 160: Priority Override Table	315
Table 161: EEPROM Command	319
Table 162: EEPROM Data	319
Table 163: AVB Command Register	320
Table 164: AVB Data Register	321
Table 165: SMI PHY Command Register	322
Table 166: SMI PHY Data Register	322
Table 167: Scratch and Misc. Register	323
Table 168: Scratch Byte 0, Register Index: 0x00 of Scratch and Misc. Control	323
Table 169: Scratch Byte 1, Register Index: 0x01 of Scratch and Misc. Control	323
Table 170: Voltage Regulator Control, Register Index: 0x06 of Scratch and Misc. Control	324
Table 171: EEE Timer Rates, Register Index: 0x0B of Scratch and Misc. Control	325

Table 172: EEE Wake Timer GE, Register Index: 0x0C of Scratch and Misc. Control	325
Table 173: EEE Assertion Timer, Register Index: 0x0D of Scratch and Misc. Control.....	325
Table 174: EEE WakeTimer, Register Index: 0x0E of Scratch and Misc. Control.....	326
Table 175: EEE TxIdle Timer, Register Index: 0x0F of Scratch and Misc. Control	326
Table 176: GPIO 0 Port Stall Vector 0, Register Index: 0x20 of Scratch and Misc. Control.....	326
Table 177: GPIO 0 Port Stall Vector 1, Register Index: 0x21 of Scratch and Misc. Control.....	327
Table 178: GPIO 1 Port Stall Vector 0, Register Index: 0x22 of Scratch and Misc. Control.....	327
Table 179: GPIO 1 Port Stall Vector 1, Register Index: 0x23 of Scratch and Misc. Control.....	328
Table 180: GPIO 2 Port Stall Vector 0, Register Index: 0x24 of Scratch and Misc. Control.....	328
Table 181: GPIO 2 Port Stall Vector 1, Register Index: 0x25 of Scratch and Misc. Control.....	329
Table 182: GPIO 3 Port Stall Vector 0, Register Index: 0x26 of Scratch and Misc. Control.....	329
Table 183: GPIO 3 Port Stall Vector 1, Register Index: 0x27 of Scratch and Misc. Control.....	330
Table 184: GPIO 4 Port Stall Vector 0, Register Index: 0x28 of Scratch and Misc. Control.....	330
Table 185: GPIO 4 Port Stall Vector 1, Register Index: 0x29 of Scratch and Misc. Control.....	331
Table 186: GPIO 5 Port Stall Vector 0, Register Index: 0x2A of Scratch and Misc. Control	331
Table 187: GPIO 5 Port Stall Vector 1, Register Index: 0x2B of Scratch and Misc. Control	332
Table 188: GPIO 6 Port Stall Vector 0, Register Index: 0x2C of Scratch and Misc. Control	332
Table 189: GPIO 6 Port Stall Vector 1, Register Index: 0x2D of Scratch and Misc. Control	333
Table 190: GPIO 7 Port Stall Vector 0, Register Index: 0x2E of Scratch and Misc. Control	333
Table 191: GPIO 7 Port Stall Vector 1, Register Index: 0x2F of Scratch and Misc. Control.....	334
Table 192: GPIO 8 Port Stall Vector 0, Register Index: 0x30 of Scratch and Misc. Control.....	334
Table 193: GPIO 8 Port Stall Vector 1, Register Index: 0x31 of Scratch and Misc. Control.....	334
Table 194: GPIO 9 Port Stall Vector 0, Register Index: 0x32 of Scratch and Misc. Control.....	335
Table 195: GPIO 9 Port Stall Vector 1, Register Index: 0x33 of Scratch and Misc. Control.....	335
Table 196: GPIO 10 Port Stall Vector 0, Register Index: 0x34 of Scratch and Misc. Control.....	336
Table 197: GPIO 10 Port Stall Vector 1, Register Index: 0x35 of Scratch and Misc. Control.....	336
Table 198: GPIO 11 Port Stall Vector 0, Register Index: 0x36 of Scratch and Misc. Control.....	336
Table 199: GPIO 11 Port Stall Vector 1, Register Index: 0x37 of Scratch and Misc. Control.....	337
Table 200: GPIO 12 Port Stall Vector 0, Register Index: 0x38 of Scratch and Misc. Control.....	337
Table 201: GPIO 12 Port Stall Vector 0, Register Index: 0x39 of Scratch and Misc. Control.....	338
Table 202: GPIO 13 Port Stall Vector 0, Register Index: 0x3A of Scratch and Misc. Control	338
Table 203: GPIO 13 Port Stall Vector 1, Register Index: 0x3B of Scratch and Misc. Control	338
Table 204: GPIO 14 Port Stall Vector 0, Register Index: 0x3C of Scratch and Misc. Control	339
Table 205: GPIO 14 Port Stall Vector 1, Register Index: 0x3D of Scratch and Misc. Control	339
Table 206: GPIO Configuration, Register Index: 0x60 of Scratch and Misc. Control	340
Table 207: GPIO Configuration, Register Index: 0x61 of Scratch and Misc. Control	342
Table 208: GPIO Direction, Register Index: 0x62 of Scratch and Misc. Control.....	344
Table 209: GPIO Direction, Register Index: 0x63 of Scratch and Misc. Control.....	344
Table 210: GPIO Data, Register Index: 0x64 of Scratch and Misc. Control	345
Table 211: GPIO Data, Register Index: 0x65 of Scratch and Misc. Control	345
Table 212: GPIO Pin Control 0, Register Index: 0x68 of Scratch and Misc. Control	346
Table 213: GPIO Pin Control 1, Register Index: 0x69 of Scratch and Misc. Control	347
Table 214: GPIO Pin Control 2, Register Index: 0x6A of Scratch and Misc. Control.....	348
Table 215: GPIO Pin Control 3, Register Index: 0x6B of Scratch and Misc. Control.....	349



Link Street®

88E6321/88E6320 Functional Specification

Table 216: GPIO Pin Control 4, Register Index: 0x6C of Scratch and Misc. Control	350
Table 217: GPIO Pin Control 5, Register Index: 0x6D of Scratch and Misc. Control	351
Table 218: GPIO Pin Control 6, Register Index: 0x6E of Scratch and Misc. Control.....	352
Table 219: GPIO Pin Control 7, Register Index: 0x6F of Scratch and Misc. Control.....	353
Table 220: CONFIG Data0, Register Index: 0x70 of Scratch and Misc. Control	354
Table 221: CONFIG Data1, Register Index: 0x71 of Scratch and Misc. Control	354
Table 222: CONFIG Data2, Register Index: 0x72 of Scratch and Misc. Control	355
Table 223: CONFIG Data3, Register Index: 0x73 of Scratch and Misc. Control	355
Table 224: Watch Dog Control Register	356
Table 225: Watch Dog Interrupt Source, Register Index: 0x00 of Watch Dog.....	357
Table 226: Watch Dog Interrupt Source, Register Index: 0x10 of Watch Dog.....	357
Table 227: Data Path Watch Dog Masks, Register Index: 0x11 of Watch Dog	358
Table 228: Data Path Watch Dog Events, Register Index: 0x12 of Watch Dog	359
Table 229: Data Path Watch Dog History, Register Index: 0x13 of Watch Dog	360
Table 230: Auto Fixing Enables, Register Index: 0x40 of Watch Dog	361
Table 231: QoS Weights Register	362
Table 232: Misc Register	363
Table 233: Misc Register	364
Table 234: PIRL Bucket Configuration Register	366
Table 235: PIRL Bucket Configuration Register	367
Table 236: PIRL Bucket Configuration Register	367
Table 237: PIRL Bucket Configuration Register	368
Table 238: PIRL Bucket Configuration Register	368
Table 239: PIRL Bucket Configuration Register	369
Table 240: PIRL Bucket Configuration Register	369
Table 241: PIRL Bucket Configuration Register	371
Table 242: PTP Port Config Register	376
Table 243: PTP Port Config Register	377
Table 244: PTP Port Config Register	378
Table 245: PTP Port Config Register	380
Table 246: PTP Port Status Register.....	381
Table 247: PTP Port Status Register.....	382
Table 248: PTP Port Status Register.....	383
Table 249: PTP Port Status Register.....	383
Table 250: PTP Port Status Register.....	384
Table 251: PTP Port Status Register.....	386
Table 252: PTP Port Status Register.....	386
Table 253: PTP Port Status Register.....	387
Table 254: PTP Port Status Register.....	387
Table 255: PTP Port Status Register.....	389
Table 256: PTP Port Status Register.....	389
Table 257: PTP Port Status Register.....	390
Table 258: PTP Port Status Register.....	390
Table 259: PTP Global Config Register, AVBPort = 0xF	393

Table 260: PTP Global Config Register, AVBPort = 0xF	393
Table 261: PTP Global Config Register, AVBPort = 0xF	394
Table 262: PTP Global Status Register, AVB = 0xF	394
Table 263: TAI Global Config Register, AVBPort = 0xE	396
Table 264: TAI Global Config Register, AVBPort = 0xE	399
Table 265: TAI Global Config Register, AVBPort = 0xE	400
Table 266: TAI Global Config Register, AVBPort = 0xE	400
Table 267: TAI Global Config Register, AVBPort = 0xE	401
Table 268: TAI Global Config Register, AVBPort = 0xE	401
Table 269: TAI Global Configuration Register, AVBPort = 0xE	402
Table 270: TAI Global Config Register, AVBPort = 0xE	403
Table 271: TAI Global Config Register, AVBPort = 0xE	404
Table 272: TAI Global Config Register, AVBPort = 0xE	404
Table 273: TAI Global Config Register, AVBPort = 0xE	405
Table 274: TAI Global Config Register, AVBPort = 0xE	405
Table 275: TAI Global Config Register, AVBPort = 0xE	405
Table 276: TAI Global Config Register, AVBPort = 0xE	406
Table 277: TAI Global Config Register, AVBPort = 0xE	406
Table 278: TAI Global Config Register, AVBPort = 0xE	406
Table 279: TAI Global Config Register, AVBPort = 0xE	407
Table 280: TAI Global Config Register, AVBPort = 0xE	407
Table 281: TAI Global Config Register, AVBPort = 0xE	407
Table 282: TAI Global Config Register, AVBPort = 0xE	408
Table 283: AVB Policy Register	410
Table 284: AVB Policy Global Clock Register, AVBPort = 0xF	414
Table 285: AVB Policy Global Legacy Register, AVBPort = 0xF	415
Table 286: AVB Policy Global Limit Register, AVBPort = 0xF	415
Table 287: AVB Policy Global OUI Register Byte 0 & 1, AVBPort = 0xF	416
Table 288: AVB Policy Global OUI Register Byte 2, AVBPort = 0xF	416
Table 289: QavPort Config Register	418
Table 290: QavPort Config Register	418
Table 291: QavPort Config Register	419
Table 292: QavPort Config Register	419
Table 293: QavPort Config Register	419
Table 294: QavPort Config Register	420
Table 295: QavPort Config Register	420
Table 296: QavPort Config Register	420
Table 297: Qav Global Config Register, AVBPort = 0xF	422
Table 298: Qav Global Status Register, AVBPort = 0xF	423
Table 299: Qav Global Status Register, AVBPort = 0xF	423
Table 300: Qav Global Status Register, AVBPort = 0xF	424
Table 301: Qav Global Status Register, AVBPort = 0xF	424
Table 302: TCAM Operation Register	426
Table 303: Keys Register 1	429



Link Street®

88E6321/88E6320 Functional Specification

Table 304: Keys Register 2	430
Table 305: Keys Register 3	431
Table 306: Keys Register 4	432
Table 307: Match Data Register 1	433
Table 308: Match Data Register 2	433
Table 309: TCAM Operation Register	435
Table 310: Match Data Register 23	435
Table 311: Match Data Register 24	435
Table 312: TCAM Operation Register	437
Table 313: Action Register 1	437
Table 314: Action Register 2	438
Table 315: Action Register 3	439
Table 316: Action Register 4	440
Table 317: TCAM Debug Register	442
Table 318: TCAM Debug Register	442
Table 319: Register Types	443
Table 320: Device Register Map Summary - Page 0 - Page 7	445
Table 321: PHY Register Map	446
Table 322: Copper Control Register	448
Table 323: Copper Status Register	450
Table 324: PHY Identifier 1	451
Table 325: PHY Identifier 2	451
Table 326: Copper Auto-Negotiation Advertisement Register	452
Table 327: Copper Link Partner Ability Register - Base Page	455
Table 328: Copper Auto-Negotiation Expansion Register	456
Table 329: Copper Next Page Transmit Register	456
Table 330: Copper Link Partner Next Page Register	457
Table 331: 1000BASE-T Control Register	457
Table 332: 1000BASE-T Status Register	458
Table 333: XMDIO MMD Access Control Register	459
Table 334: XMDIO MMD Access Address/Data Register	459
Table 335: Extended Status Register	460
Table 336: Copper Specific Control Register 1	460
Table 337: Copper Specific Status Register 1	462
Table 338: Copper Specific Interrupt Enable Register	464
Table 339: Copper Interrupt Status Register	465
Table 340: Copper Specific Control Register 2	466
Table 341: Copper Specific Receive Error Counter Register	466
Table 342: Page Address	467
Table 343: Global Interrupt Status	467
Table 344: Copper Specific Control Register 3	468
Table 345: MAC Specific Control Register 1	470
Table 346: MAC Specific Interrupt Enable Register	471
Table 347: MAC Specific Status Register	472

Table 348: Copper RX_ER Byte Capture	472
Table 349: MAC Specific Control Register 2	473
Table 350: Advanced VCT™ TX to MDI[0] Rx Coupling	474
Table 351: Advanced VCT™ TX to MDI[1] Rx Coupling	475
Table 352: Advanced VCT™ TX to MDI[2] Rx Coupling	476
Table 353: Advanced VCT™ TX to MDI[3] Rx Coupling	477
Table 354: 1000BASE-T Pair Skew Register	478
Table 355: 1000BASE-T Pair Swap and Polarity	478
Table 356: Advanced VCT™ Control	479
Table 357: Advanced VCT™ Sample Point Distance.....	479
Table 358: Advanced VCT™ Cross Pair Positive Threshold.....	480
Table 359: Advanced VCT Same Pair Impedance Positive Threshold 0 and 1.....	480
Table 360: Advanced VCT™ Same Pair Impedance Positive Threshold 2 and 3.....	480
Table 361: Advanced VCT™ Same Pair Impedance Positive Threshold 4 and Transmit Pulse Control	481
Table 362: Copper Port Packet Generation.....	481
Table 363: Copper Port CRC Counters	482
Table 364: Checker Control.....	482
Table 365: Misc Test	483
Table 366: Temperature Sensor	483
Table 367: PHY Cable Diagnostics Pair 0 Length	484
Table 368: PHY Cable Diagnostics Pair 1 Length	484
Table 369: PHY Cable Diagnostics Pair 2 Length	484
Table 370: PHY Cable Diagnostics Pair 3 Length	484
Table 371: PHY Cable Diagnostics Results	485
Table 372: PHY Cable Diagnostics Control	486
Table 373: Advanced VCT™ Cross Pair Negative Threshold	486
Table 374: Advanced VCT Same Pair Impedance Negative Threshold 0 and 1	486
Table 375: Advanced VCT Same Pair Impedance Negative Threshold 2 and 3	487
Table 376: Advanced VCT Same Pair Impedance Negative Threshold 4	487
Table 377: Register Map	488
Table 378: PCS control 1 register.....	489
Table 379: PCS status 1 register.....	489
Table 380: PCS EEE capability register	490
Table 381: PCS EEE wake error counter	490
Table 382: EEE advertisement register	491
Table 383: EEE Link partner advertisement register	491
Table 384: Register Types.....	492
Table 385: Fiber/SERDES Register Map Summary - Page 1.....	493
Table 386: Fiber/SERDES Registers - Register Map	494
Table 387: Fiber Control Register.....	495
Table 388: Fiber Status Register	497
Table 389: PHY Identifier.....	498
Table 390: PHY Identifier.....	498
Table 391: Fiber Auto-Negotiation Advertisement Register - 1000BASE-X Mode (Register 16_1.1:0 = 01) ..	499



Link Street®

88E6321/88E6320 Functional Specification

Table 392: Fiber Auto-Negotiation Advertisement Register - SGMII (System mode) (Register 16_1.1:0 = 10).....	501
Table 393: Fiber Auto-Negotiation Advertisement Register - SGMII (Media mode) (Register 16_1.1:0 = 11).....	501
Table 394: Fiber Link Partner Ability Register - 1000BASE-X Mode (Register 16_1.1:0 = 01).....	502
Table 395: Fiber Link Partner Ability Register - SGMII (System mode) (Register 16_1.1:0 = 10).....	503
Table 396: Fiber Link Partner Ability Register - SGMII (Media mode) (Register 16_1.1:0 = 11).....	503
Table 397: Fiber Auto-Negotiation Expansion Register.....	504
Table 398: Fiber Next Page Transmit Register	504
Table 399: Fiber Link Partner Next Page Register.....	505
Table 400: Extended Status Register	505
Table 401: Fiber Specific Control Register 1	505
Table 402: Fiber Specific Status Register	507
Table 403: Fiber Interrupt Enable Register.....	508
Table 404: Fiber Interrupt Status Register.....	509
Table 405: Fiber Receive Error Counter Register	509
Table 406: Fiber Page Register	509
Table 407: PRBS Control	510
Table 408: PRBS Error Counter LSB	510
Table 409: PRBS Error Counter MSB	510
Table 410: Fiber Specific Control Register 2	511

List of Figures

Figure 1:	Switch Data Flow	24
Figure 2:	Fields of the Frame Examined for Layer 2 PCL.....	44
Figure 3:	IPv4 DHCP Option 82 Frame Format	46
Figure 4:	IPv6 DHCP Option 82 Frame Format	46
Figure 5:	Switch Operation with Port VLANS Disabled.....	47
Figure 6:	Switch Operation with a Typical Router VLAN Configuration	48
Figure 7:	Switch Operation with another Example VLAN Configuration	49
Figure 8:	IEEE Tag Frame Format.....	50
Figure 9:	Relationship between VTU, STU, and ATU	53
Figure 10:	ARP Mirror Format.....	56
Figure 11:	IPv4 IGMP Snoop Format.....	57
Figure 12:	IPv6 MLD Snoop Format	57
Figure 13:	IEEE Tag Format	58
Figure 14:	Port's IEEE PRI Mapping.....	59
Figure 15:	IPv4 Priority Frame Format	59
Figure 16:	IPv6 Priority Frame Format	59
Figure 17:	Port's IP Pri Mapping	60
Figure 18:	Port's Default PRI Mapping	60
Figure 19:	Color blind Leaky Bucket for Rate Limiting	65
Figure 20:	Data Rate Limiting Example	66
Figure 21:	Priority Based Rate Limiting Example	67
Figure 22:	Switch Queues.....	70
Figure 23:	IEEE Tag Format	76
Figure 24:	Provider vs. Normal Data Planes.....	81
Figure 25:	Provider Tag Format.....	82
Figure 26:	Provider S-PRI Remapping	84
Figure 27:	Provider IP PRI Mapping	84
Figure 28:	Forward DSA Tag Format.....	87
Figure 29:	To_CPU DSA Tag Format	90
Figure 30:	From_CPU DSA Tag Format	92
Figure 31:	To_Sniffer DSA Tag Format	94
Figure 32:	Cross-chip Port VLAN Table Formats	98
Figure 33:	Ether Type DSA Tag Format	102
Figure 34:	Normal and Provider vs. Control Data Planes	104
Figure 35:	Ingress Header Format	109
Figure 36:	Egress Header Format - Header Type = 00 (Original Header).....	110
Figure 37:	Egress Header Format - Header Type = 01 (Single chip MGMT Header).....	111
Figure 38:	Trunk Mask Load Balancing Table - Example	116
Figure 39:	PTP Common Header Format	119
Figure 40:	Format of a TCAM Entry	122
Figure 41:	Format of a VTU Entry	131



Link Street®

88E6321/88E6320 Functional Specification

Figure 42: Format of an STU Entry	136
Figure 43: Format of an ATU Entry	142
Figure 44: Remote Management DSA Tag Request Format	151
Figure 45: Remote Management DSA Tag Response Format	153
Figure 46: Remote Management Generic Layer 3 Request Format	155
Figure 47: Remote Management Generic Layer 3 Response Format	156
Figure 48: Remote Management ATU Dump Response Format	158
Figure 49: Remote Management MIB Dump Response Format	159
Figure 50: Remote Management Register Read/Write Response Format	161
Figure 51: Remote Management Error Response Format	163
Figure 52: MAC Interface Loopback Diagram	167
Figure 53: Line Loopback Data Path	168
Figure 54: Loopback Stub (Top View with Tab up)	169
Figure 55: Test Setup for 10/100/1000 Mbps Modes using an External Loopback Stub	169
Figure 56: Cable Fault Distance Trend Line	180
Figure 57: First Peak Example	181
Figure 58: CML I/Os	190
Figure 59: Synchronous SERDES Loopback Diagram	192
Figure 60: Typical MDC/MDIO Read Operation	198
Figure 61: Typical MDC/MDIO Write Operation	198
Figure 62: Device Register Map	214
Figure 63: Per Port Register Bit Map	215
Figure 64: Global 1 Register Bit Map	266
Figure 65: Global 2 Register bit Map (Device Addr 0x1C)	293
Figure 66: PIRL Register bit Map (from Global 2 offsets 0x09 and 0x0A)	365
Figure 67: PTP Global Configuration Data Structure Registers	375
Figure 68: PTP Global Register bit Map (Global 2 offset 0x16 and 0x17 w/AVBBlock = 0x0 & AVBPort = 0xF)	392
Figure 69: PTP TAI Global Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x0 & AVBPort = 0xE)	395
Figure 70: AVB Policy Port Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x1 & AVBPort = 6:0)	409
Figure 71: AVB Policy Global Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x1 & AVBPort = 0xF)	413
Figure 72: Qav Port Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x2 & AVBPort = 6:0)	417
Figure 73: QavGlobal Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x2 & AVBPort = 0xF)	421
Figure 74: Global 3 Register bit Map for TCAM Page 0 (Device Addr 0x1D)	425
Figure 75: Global 3 Register bit Map for TCAM Page 1 (Device Addr 0x1D)	434
Figure 76: Global 3 Register bit Map for TCAM Page 2 (Device Addr 0x1D)	436
Figure 77: EEPROM Data Format	513

1

Switch Core Functional Description

The device has been designed for many different applications. For flexibility, and to facilitate learning how to use the switch, all switch ports have been designed with identical capabilities as far as the switch core is concerned¹. At the same time, the physical port speed options and connections to the outside world are different depending upon the port number (see Applications Examples in 88E6321/88E6320 Functional Specification Datasheet, Part 1 of 3: Overview, Pinout, Applications, Mechanical and Electrical Specifications for details.).

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

- Basic Switch Functions – those functions that are common to all Frame Modes ([Section 2](#)).
- Normal Network Frame Mode – for IEEE standard untagged and tagged frames or for ‘customer’ ports on switches with at least one ‘provider’ port ([Section 3](#)).
- Provider Frame Mode – for IEEE standard ‘provider’ ports ([Section 4](#)).
- Distributed Switch Architecture (DSA) Frame Mode – for multiple chip switch fabric extensions or for connections to a switch management CPU and/or Router CPU ([Section 5](#)).

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

- Normal
- Provider
- DSA

Where two DSA options are supported:

- Classic
- Ether type

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

1. There is one exception: Remote Management ([Section 8](#)) is not supported on all ports for security reasons.

2

Basic Switch Functions

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

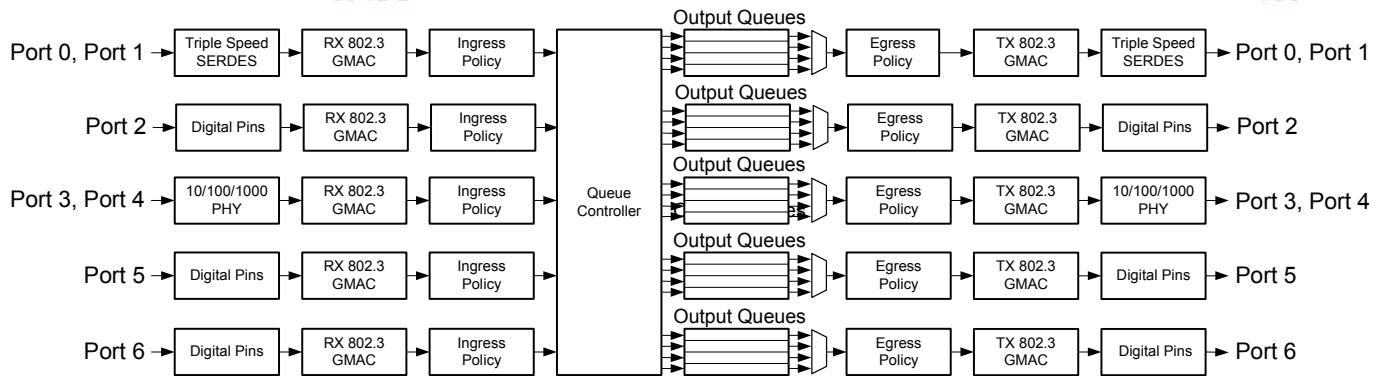
2.1

Physical Switch Data Flow

The device accepts Ethernet frames and either discards them or transmits them out of one or more of the switch's ports. The decision on what to do with each frame is just one of the many tasks handled inside the switch. [Figure 1](#) shows the data path inside the switch along with the major functional blocks that process the frame as it travels through the device. Each of these blocks along with their register-controllable options and policy is described in the following sections.

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

Figure 1: Switch Data Flow



2.2

Physical Interface

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



Note

Device features are discussed with references to the registers that control the features. The registers in the switch device are organized into three groups called Port, Global 1, Global 2, and Global 3 with an additional group used to access the PHYs called PHY. Each of these groups support up to 32 16-bit registers and each port has its own set of 32 Port registers. A specific register out of the 32 in a group can be referred to by the term 'offset'. For example, the Port Control register is referenced as Port offset 0x04 as it appears in the Port device address space at register address 0x04. See [Section 10](#) for details on the registers.

2.3

Media Access Controllers (MAC)

The devices contain seven MACs. These MACs perform all of the functions in the 802.3 protocol including frame Formatting, frame stripping, CRC checking, CSMA/CD enforcement, and collision handling. Each MAC supports 1 Gbps operation in full-duplex mode only and 10/100 Mbps operation in full-duplex or half-duplex mode.

The MAC receive block checks incoming packets and discards those with CRC errors, those with alignment errors, short packets (less than 64 bytes), long packets (more than 1522 bytes)¹ in non-Jumbo mode, and Jumbo packets (10240 bytes). Each MAC constantly monitors its receive lines waiting for preamble bytes followed by the Start of Frame Delimiter (SFD). The first six bytes after the SFD are used as the packet's Destination Address (DA)² and the next six bytes after that are used as the packet's Source Address (SA). These two addresses are fundamental to the operation of the switch (see [Section 2.4](#) for more information). The next two to sixty bytes are examined and may be used for QoS (Quality of Service) or policy decisions made by the switch (see [Section 3](#) for more information). The last four bytes of the packet contain the packet's Frame Check Sequence (FCS). The FCS must meet the IEEE 802.3 CRC-32 requirements for the packet or it will be discarded.

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

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

1. A maximum frame size of 10240 bytes is supported by setting the MaxFrameSize bit in the Switch Port register (Port Control 2, offset 0x08).
2. The first six bytes of a frame are processed as the frame's DA unless the Marvell® Header mode is enabled on the port (Port offset 0x04). If the Marvell Header mode is enabled the first two bytes are processed as the Marvell Header and the next six bytes are processed as the frame's DA.



2.3.1

Backoff

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

2.3.2

Half-duplex Flow Control

Half-duplex flow control is used to throttle the throughput rate of an end station to avoid dropping packets during network congestion. It is enabled on all half-duplex ports via the EE_CLK/C0_LED/FLOW pin (see 88E6321/88E6320 Datasheet for details). Flow control can be enabled or disabled on any particular port by forcing the port's Flow Control mode (FCValue and ForcedFC in the PCS Control Register, Port offset 0x01). The device uses a mixed carrier assertion and collision based scheme to perform half-duplex flow control. When the free buffer space is almost empty, the MAC issues carrier and/or collision which prevents further incoming packets. Only the ports that are involved in the congestion are flow controlled. If the half-duplex flow control mode is not set and there is no packet buffer space available, the incoming packet is discarded.

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

2.3.3

Full-duplex Flow Control

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

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

Full-duplex flow control is enabled on all full-duplex ports via the EE_CLK/C0_LED/FLOW pin (see 88E6321/88E6320 Datasheet for details), if Auto-Negotiation is enabled on the port, and if the link partner 'advertises' that it supports Pause during Auto-Negotiation. Basically, full-duplex flow control is automatically enabled on a port if:

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

The EE_CLK/C0_LED/FLOW pin (at the rising edge of RESETn) determines the initial flow control advertisement bit setting in the PHYs of this device. The value of this pin is moved to external PHYs by the PPU.

When flow control is enabled using the EE_CLK/C0_LED/FLOW device pin, it is enabled for all ports of the same type (i.e., on all half-duplex ports or on all full-duplex ports that have a flow-controllable link partner). It may be required to have flow control enabled on only one or two ports and disable flow control on all other ports. In this case, flow control should be disabled via the

EE_CLK/C0_LED/FLOW device pin, which will disable flow control on all the ports. The ports chosen to have flow control enabled can then be configured to advertise flow control. This can be done by using the SMI PHY Command and Data Registers - Global 2 offset 0x18 and 0x19.

In full-duplex mode, the device's MACs support flow control as defined in the IEEE 802.3 standard. This flow control mechanism enables stopping and restarting packet transmission at the remote node. The basic mechanism for performing full-duplex flow control is via a Pause frame. The format of the Pause frame is shown in [Table 2](#).

Table 2: Pause Frame Format

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

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

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

The Source Address of a received Pause frame is not learned¹ since it may not represent the Source Address of the transmitting port. This is generally the case if the link partner is an unmanaged switch. The Source Address of transmitted Pause frames can be configured (see switch MAC address register, Global 2 offset 0x0D). A single fixed Source Address can be used for all ports, or a unique Source Address per port can be selected by changing the value of the DiffAddr bit in the switch MAC Address register.

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

2.3.4

Forcing Flow Control in the MAC

[Section 2.3.3](#) describes the IEEE defined flow control mechanism, which requires auto-negotiation with a link partner. Some ports may not have a PHY attached, or there may be a PHY attached without auto-negotiation. In this case, flow control can be enabled or disabled by forcing the port's Flow Control mode (FCValue and ForcedFC in bit in the port's PCS Control Register, Port offset 0x01). Forcing flow control in this way will instruct the port's MAC to transmit Pause frames when needed and act on received Pause frames. It does not change the advertisement bits in the port's PHY².

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

-
1. See Automatic Address Learning in [Section 2.4.3](#).
 2. In this case the port's link partner may not be supporting Pause frames because it does not see from the PHY that this port is advertising it wants to support Pause.



Link Street®

88E6321/88E6320 Functional Specification

2.3.5

Jamming Control - Egress Limit

Perfect flow control, i.e., no packet loss, (full- or half-duplex) can cause network problems. A potential problem can occur between two switch boxes that simultaneously Pause each other off at exactly the same time such that neither can drain their full buffers. This very rare, but possible, situation can cause a dead-lock on the link between the two switch boxes. It is easily solved by limiting the number of back-to-back Pause refreshes a port can transmit and thus the maximum time the link partner can be stalled, allowing the dead-lock to clear. The Port's LimitOut register (Port offset 0x02) controls the number of maximum Pause times that a port can stall its link partner. The range of the LimitOut register is large enough to ensure zero packet loss under normal, or even extreme, network congestion situations while at the same time ensuring a dead-lock situation does not occur.

**Note**

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

2.3.6

Jamming Control - Ingress Limit

When flow control is enabled on a port, it can be stalled by its link partner such that the port can never transmit any frames. This could be a result of the problem described above ([Section 2.3.5](#)) or it could be a DoS attack (Denial of Service). The port's Limithn register (Port offset 0x03) can be used to limit how long a port's egress queue can be jammed. Once the programmed limit is reached, flow control will be forced off on the port and an interrupt generated to the CPU (if enabled – Global 2 offset 0x00 and 0x01). Software can determine which port reached the limit by examining the ports flow control forcing bits. Flow control will be forced off on any port whose limit was reached (ForcedFC will = 1 and FCValue will = 0 in Port offset 0x01). Software can re-enable flow control on the port changing the value of these bits (by clearing the port's ForcedFC bit to zero).

If the port is in full-duplex mode the automatic disabling of flow control on the port will allow frames to egress the port once the last Pause time has expired. But if the port is in half-duplex mode, constant collisions from the link partner can still prevent frames to egress the jammed port. For this reason, the Jam Limit interrupt will be activated on half-duplex ports even if flow control is disabled on these ports. Software can take action to either enable DiscardExcessive (see [Section 2.3.1](#) or Global 1 offset 0x04) or to Disable or Block the port (see Port States in Port offset 0x04). In either case, the goal is to free up the jammed buffers for other ports to use (see [Section 3.6](#)).

2.3.7

Forcing Link, Speed, and Duplex in the MAC

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

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

- When no PHY is connected to the port. This includes ports that connect to a CPU (typically using a digital interface like MII or GMII).

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

2.3.8

MAC Based RMON/Statistics Counters

The MAC Based Statistics Counter logic maintains a set of 28, 32-bit counters and two 64-bit counters per port, that enable the user to monitor network performance remotely and to support RMON groups 1, 2, 3, and 9. These counters provide a set of Ethernet statistics for frames received on ingress and transmitted on egress. Switch Policy Statistics counters are also supported. See [Section 2.3.9](#).

The counters are designed to support:

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

The complete description of each of the counters is contained in [Table 3](#) and [Table 4](#).

All CPU register interfaces are slow compared with the speed of Gigabit or even Fast Ethernet frames. For this reason all the RMON counter data associated with a port can be placed into an Ethernet frame and transmitted to the CPU (or other device). Two options are supported, a MIB Dump and a MIB Dump and Clear. See Remote Management described in [Section 8](#).

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

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

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

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



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

Note



Link Street®

88E6321/88E6320 Functional Specification

Table 3: Ingress Statistics Counters

Name	Offset Address	Description
Set 1		
InGoodOctetsLo	0x00	The lower 32-bits of the 64-bit InGoodOctets counter. The sum of lengths of all good Ethernet frames received, that is frames that are not bad frames.
InGoodOctetsHi	0x01	The upper 32-bits of the 64-bit InGoodOctets counter. See description above.
InBadOctets	0x02	The sum of lengths of all bad Ethernet frames received.
Set 2		
InUnicast	0x04	The number of good frames received that have a Unicast destination MAC address.
InBroadcasts	0x06	The number of good frames received that have a Broadcast destination MAC address.
InMulticasts	0x07	The number of good frames received that have a Multicast destination MAC address. NOTE: This does not include frames counted in InPause nor does it include frames counted in InBroadcasts.
InPause	0x16	The number of good frames received that have a Pause destination MAC address.
Set 3		
InUndersize	0x18	Total frames received with a length of less than 64 octets but with a valid FCS.
InFragments	0x19	Total frames received with a length of less than 64 octets and an invalid FCS.
InOversize	0x1A	Total frames received with a length of more than MaxSize octets but with a valid FCS.
InJabber	0x1B	Total frames received with a length of more than MaxSize octets but with an invalid FCS.
InRxErr	0x1C	Total frames received with an RxErr signal from the PHY.
InFCSErr	0x1D	Total frames received with a CRC error not counted in InFragments, InJabber or InRxErr.
Set 4		
64Octets	0x08	These counters can be Ingress Only, Egress Only, or both Total frames received (and/or transmitted) with a length of exactly 64 octets, including those with errors.
65to127Octets	0x09	Total frames received (and/or transmitted) with a length of between 65 and 127 octets inclusive, including those with errors.
128to255Octets	0x0A	Total frames received (and/or transmitted) with a length of between 128 and 255 octets inclusive, including those with errors.
256to511Octets	0x0B	Total frames received (and/or transmitted) with a length of between 256 and 511 octets inclusive, including those with errors.

Table 3: Ingress Statistics Counters

Name	Offset Address	Description
512to1023Octets	0x0C	Total frames received (and/or transmitted) with a length of between 512 and 1023 octets inclusive, including those with errors.
1024toMaxOctets	0x0D	Total frames received (and/or transmitted) with a length of between 1024 and MaxSize ¹ octets inclusive, including those with errors.

1. MaxSize is 1522 in non-Jumbo mode and 10240 for Jumbo packets for non-tagged frames and for tagged frames if MaxFrameSize = 0 or MaxSize = 10240 if MaxFrameSize = 1 (Port Control 2 offset 0x08).



Link Street®

88E6321/88E6320 Functional Specification

Table 4: Egress Statistics Counters

Name	Offset Address	Description
Set 5		
OutOctetsLo	0x0E	The lower 32-bits of the 64-bit OutOctets counter. The sum of lengths of all Ethernet frames sent from this MAC.
OutOctetsHi	0x0F	The upper 32-bits of the 64-bit OutOctets counter. See description above.
Set 6		
OutUnicast	0x10	The number of frames sent that have a Unicast destination MAC address.
OutBroadcasts	0x13	The number of good frames sent that have a Broadcast destination MAC address.
OutMulticasts	0x12	The number of good frames sent that have a Multicast destination MAC address. NOTE: This does not include frames counted in OutPause nor does it include frames counted in OutBroadcasts.
OutPause	0x15	The number of Flow Control frames sent.
Set 7		
Deferred	0x05	The total number of successfully transmitted frames that experienced no collisions but are delayed because the medium was busy during the first attempt. This counter is applicable in half-duplex only.
Collisions	0x1E	The number of collision events seen by the MAC not including those counted in Single, Multiple, Excessive, or Late. This counter is applicable in half-duplex only.
Single	0x14	The total number of successfully transmitted frames that experienced exactly one collision. This counter is applicable in half-duplex only.
Multiple	0x17	The total number of successfully transmitted frames that experienced more than one collision. This counter is applicable in half-duplex only.
Excessive	0x11	The number of frames dropped in the transmit MAC because the frame experienced 16 consecutive collisions. This counter is applicable in half-duplex only and only if DiscardExcessive is a one (in Switch Global Control - global offset 0x04).
Late	0x1F	The number of times a collision is detected later than 512 bits-times into the transmission of a frame. This counter is applicable in half-duplex only.
OutFCSErr	0x03	The number of frames transmitted with an invalid FCS. Whenever a frame is modified during transmission (e.g., to add or remove a tag) the frame's original FCS is inspected before a new FCS is added to a modified frame. If the original FCS is invalid, the new FCS is made invalid too and this counter is incremented.
Set 4		These counters can be Ingress Only, Egress Only, or both
64Octets	0x08	Total frames transmitted (and/or received) with a length of exactly 64 octets, including those with errors.

Table 4: Egress Statistics Counters

Name	Offset Address	Description
65to127Octets	0x09	Total frames transmitted (and/or received) with a length of between 65 and 127 octets inclusive, including those with errors.
128to255Octets	0x0A	Total frames transmitted (and/or received) with a length of between 128 and 255 octets inclusive, including those with errors.
256to511Octets	0x0B	Total frames transmitted (and/or received) with a length of between 256 and 511 octets inclusive, including those with errors.
512to1023Octets	0x0C	Total frames transmitted (and/or received) with a length of between 512 and 1023 octets inclusive, including those with errors.
1024toMaxOctets	0x0D	Total frames transmitted (and/or received) with a length of between 1024 and MaxSize ¹ octets inclusive, including those with errors.

1. MaxSize is 1522 in non-Jumbo mode and 10240 for Jumbo packets for non-tagged frames and for tagged frames if MaxFrameSize = 0 or MaxSize = 10240 if MaxFrameSize = 1 (Port Control 2 offset 0x08).

2.3.9 Policy Based RMON/Statistics Counters

The device maintains a set of policy counters (one 32-bit and two 16-bit) per port that enable the user to monitor network performance by seeing where good frames have been dropped by the switch (bad frames that are dropped are counted in the MAC based counters – [Section 2.3.8](#)). Some frames are dropped due to switch policy and others are due to excessive congestion in the switch.

The policy counters are:

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

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

All CPU register interfaces are slow compared with the speed of Gigabit or even Fast Ethernet frames. For this reason, all the RMON counter data associated with a port can be placed into an Ethernet frame and transmitted to the CPU (or other device). Two options are supported, a MIB Dump and a MIB Dump and Clear. See Remote Management described in [Section 8](#).



Link Street®

88E6321/88E6320 Functional Specification

2.4

Basic Switch Operation

The switch portion of the device receives good packets from the MACs, processes them, and forwards them to the appropriate MACs for transmission. The primary task of the switch is to process frames and this activity involves the following blocks shown in [Figure 1](#).

- Ingress Policy ([Section 3.1](#))
- Queue Controller ([Section 3.6](#))
- Output Queues ([Section 3.6.5.2](#))
- Egress Policy ([Section 3.8](#))

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

The normal packet flow and processing through a switch involves learning how to switch packets to the correct MACs, and only to the correct ones. The switch learns what port an end station is connected to by remembering each packet's Source Address¹ along with the port number on which the packet arrived. Once a MAC address/port number mapping is learned, all future packets directed to that end station's MAC address (as defined in a frame's Destination Address field) are directed to the learned port number only. If a packet is directed to a new, currently unlearned, MAC address, the packet will be transmitted out of all the ports², except for the one on which it arrived³. This ensures that the packet is received by the correct end station (if it exists) and when the end station responds back its address is learned by the switch for the next series of packets.

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

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

2.4.1

Lookup Engine

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

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

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

1. The SA on switch management frames ([Section 6.1](#)) are not learned. This includes the IEEE Pause frame.
2. VLANs modify this operation – see [Section 3.2](#) and [Section 3.2.2](#).
3. The device can be configured to transmit frames out the port they came in on – see [Section 3.3.1](#).

2.4.2

Address Searching or Translation

The address search engine is used to search the address database to get the output port number(s), called the Destination Port Vector (DPV), for each frame's destination address so that it can switch the frame instead of flooding¹ it. It arbitrates destination address lookup requests from the ports and grants one lookup at a time. The address is hashed and then data is read from the SRAM table, looking for a MAC address match. Four addresses can be stored at each hash location. If a match is found, the Address Translation Unit (ATU) returns the Destination Port Vector (DPV) to the Ingress Policy block where it may be modified² before the packet is queued to the output ports. If the found entry contains a Trunk ID, the Trunk ID is converted to a DPV using the Trunk Mapping Table (Global 2, offset 0x08). If no MAC address match is found the Ingress Policy block uses a unique default DPV for each ingress port³, which typically floods the frame. If the destination address in the frame is a multicast address or broadcast address, the address is searched⁴ in the same way as a unicast address and the frame is processed identically. This feature is used for multicast filtering. Multiple separate address databases are supported in the device. The database that is searched is controlled by the port's default Forwarding Information Database (FID in the Port Based VLAN Map register, Port offset 0x06, and the Port Control 1 register, Port offset 0x05) or the one assigned to the frame by the VTU ([Section 7.2](#)) based on the VID assigned to the frame during ingress ([Section 3.2.2](#)). MAC addresses that are not members of the port's or frame's FID cannot be found.

2.4.3

Automatic Address Learning

The address learning engine is used to learn the source address of ingressing frames. Up to 8192 MAC address/port number mappings can be stored in the address database (see [Section 2.4.1](#)) for more information). When the source address from an input frame can not be found in the address database, the ATU enters the self-learning mode and places the new MAC address/port number mapping into the database and refreshes its Age time⁵. If the MAC address is found to be already in the database, the port information and Age associated with the entry is updated and/or refreshed. The port number/Trunk ID is updated in case the end station moved and the port number or Trunk ID needs to be corrected. The entry's Age is refreshed since the MAC address is still 'active'. This prevents the MAC address/port number mapping from being removed as being 'inactive' prematurely.

When an address is added into the database it is hashed and stored in the first empty bin found at the hashed location. If all four address bins are full, a least recently used algorithm is used for looking at each entry's Age time (its EntryState field⁶). If all four address bins have the same Age time, then the first 'non-static' bin is used (see [Section 7.3.1](#) for more information about locked or static addresses). If all four bins are 'static' the address is not learned and an ATUFull interrupt is generated (see the Switch Global Status register, Global 1 offset 0x00). The port information stored with the new MAC address is the port's Port Association Vector (PAV, Port offset 0x0B) if the source port is not a Trunk port (Trunk Port bit Port offset 0x05) or it is the port's Trunk ID (Port offset 0x05) if the source port is a Trunk port.

Multiple separate address databases are supported in the device (see [Section 2.4.8](#)). The port's Forwarding Information Database (FID in the Port Based VLAN Map register, Port offset 0x06, and the Port Control 1 register, Port offset 0x05) determines into which address database the MAC address is added if 802.1Q is disabled on the port. If 802.1Q is enabled on the port the FID associated with the frame's VID (VLAN ID field of Tagged frames, see [Section 3.2.2.1](#)) determines the address database into which the MAC address is stored. The same MAC address can be learned multiple times with different port mappings if different FID values are used.

1. Flooding refers to the action of sending frames out all the ports of the switch except for the port the frame came in on.
2. The DPV returned from the ATU may be modified by the VLANTable data, VTU results, the Trunk Mask Table, and/or other filters.
3. The default DPV for each port is the list of ports that can egress multicast frames, if the frame is multicast or the list of ports that can egress unicast frames, if the frame is unicast (Egress Floods in Port Control register, Port offset 0x04). Broadcast frames are considered multicast frames unless Flood BC is set to a one (Global 2, offset 0x05).
4. Multicast addresses cannot be auto learned. Multicast addresses must be loaded manually with a CPU or EEPROM.
5. The Age time on a MAC Address entry is refreshed by setting its EntryState field to 0x7- see [Table 36](#).
6. The EntryState field is described in [Section 7.3.1](#).



Link Street®

88E6321/88E6320 Functional Specification

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

Learning is never performed on switch management frames. This includes Pause frames ([Section 2.3.3](#)), BPDU, LAC and other management frames as long as they are determined to be MGMT frames (see Management frames - [Section 6.1](#)), and Distributed Switching Architecture (DSA) Tag frames ([Section 5](#)) with the exception of the Forward type of DSA Tag frames.

2.4.4

Hardware Address Learn Limiting

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

1. Frames containing a Source Address (SA) in the address database that is associated with the port the frame entered will be allowed to enter the port¹. This is true for all MAC addressed in the address database, including those already in the database prior to the Learn Limit being enabled, and those added by the CPU either as static or aging.
2. Frames containing an SA that is not in the address database or one that is in the address database, but is not associated with the port the frame entered, will be discarded. If the frame contains a new SA an ATU Miss Interrupt will be generated (see [Section 7.3.6](#)) if the port's Over Limit Int En bit is set to a one (Port offset 0x0C).

The learn counter will decrement by one² whenever an address associated with this port ages out of the address database (see [Section 2.4.5](#)). In this case the port's Limit Reached bit will be cleared indicating the counter is below the limit. Now if a frame with a new Source Address enters the port, the frame will be accepted and its SA will be learned because the limit counter is less than the Learn Limit.

Only automatic operations affect the port's learn counter (LearnCtr). CPU ATU operations such Load and Purge (see [Section 7.3.3](#)) do not effect the port's learn counter. The only exceptions to this are the ATU Flush All entries and the ATU Flush All Non-Static entries commands. Since both of these operations clear out all non-static entries, all port's learn counters are reinitialized to zero.

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

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

1. Disable learning on the port. Either clear the port's PAV (Port offset 0x0B) or set the port's LearnDisable bit (Port offset 0x06).
2. Clear out all addresses associated with this port in the address database. Either issue an ATU Flush All Non-Static or do an ATU Move Non-Static to port 0xF (Global 1 offset 0x0B).
3. Clear the port's LearnLimit to zero to reinitialize the port's LearnCtr (this is not needed if the ATU Flush All Non-Static operation was used above).
4. Set the port's LearnLimit to the new value.

1. The entry's Age will be refreshed as well if it is not static (see [Section 2.4.5](#))
2. The decrement is clipped at zero to cover the case where addresses were already present in the address database, associated with this port, prior to the Learn Limit being enabled.

5. Re-enable learning on the port by reversing what was done in step 1 above.



Note

- Hardware Address Learn Limiting requires that Learn2All (Global 1 offset 0x0A) must be set to a one and that Locked Port is cleared to zero on the port (Port offset 0x0B) and that the port is not a member of a Trunk (Trunk Port is cleared to zero in Port offset 0x05).
- If either the source port and/or the destination port of a station move¹ has hardware Address Learn Limiting enabled, the station move will not take place. This is a self correcting situation as the station move will take place as soon as the MAC address of the station move ages out ([Section 2.4.5](#)) as it will if the station actually moved.

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

2.4.5 Automatic Address Aging

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

The device runs the address aging process continuously (unless disabled by setting the AgeTime field to zeros). Aging is accomplished by a periodic sweeping of the address database. The speed of these sweeps determines the aging time. On each aging sweep of the database, the ATU reads each valid entry and updates its age time by decrementing its EntryState field¹ (as long as the entry is not static – see [Section 7.3.1](#)). When the EntryState field reaches zero, the entry is considered invalid and purged from the database. The EntryState field will not decrement past 0x1 (i.e., it will not be automatically purged) if the port's HoldAt1 bit is set (Port offset 0x0B). HoldAt1 is intended to be used with CPU directed Address Learning ([Section 2.4.6](#)).

A new or just refreshed unicast MAC address has an EntryState value of 0x7 (see [Section 2.4.3](#)). A purged or invalid entry has an EntryState value of 0x0. The values from 0x6 to 0x1 indicate the Age time on the unicast MAC address with 0x1 being the oldest. This scheme results in seven age states on an entry allowing the Address Learning's least recently used replacement process ([Section 2.4.3](#)) to be more precise. An address is purged from the database within 1/7th of the programmed AgeTime value making the address's lifetime interval in the database more accurate as well.

2.4.6 CPU Directed Address Learning and Purging

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

The CPU can retrieve the source MAC address and the source port information of the SA Miss Violation by issuing an ATU Get/Clear Violation Data ATUOp ([Section 7.3.6](#)). The CPU then decides if the address should be placed into the address database or not. If it should be, the CPU issues a Load ATUOp ([Section 7.3.3](#)).

1. The EntryState field is described in [Section 7.3.1](#).



Link Street®

88E6321/88E6320 Functional Specification

If the CPU loads the new address as 'non-static', the entry stays in the address database until it ages out. Its age time is determined by the loaded EntryState value¹, as addresses on Locked ports do not have their age time refreshed, unless the port's RefreshLocked bit is set to a one (Port offset 0x0B). In this case, addresses already present in the address database will be automatically refreshed (i.e., get their EntryState set to 0x7) as long as the association on the address is not changing (i.e., as long as it is not a station move). Learn2All message frames are generated on these automatic refreshes if the Learn2All bit is set to a one (Global 1 offset 0x0A).

The CPU receives no new interrupts from non-static addresses until they age out or are purged out by the CPU, unless the ATUAgeIntEn bit is set to a one (Global 2 offset 0x5) and/or unless the port's IntOnAgeOut bit is set to a one (Port offset 0x0B). If the global ATUAgeInt is enabled the CPU will receive an ATU Miss Violation if the EntryState found in the address database on the ingressing frame's SA is less than 0x4 (and the port's RefreshLocked bit is cleared to zero). If the port's IntOnAgeOut (interrupt on age out) is enabled, the CPU will receive an Age Out Violation whenever any address associated with the port² is at an EntryState of 0x1 when aging starts to process the entry. The entry will then either be aged out of the address database, or its EntryState value will be held a 0x1 if the port's HoldAt1 bit is set (Port offset 0x0B). The Hold at 1 feature requires that the CPU to purge the entry from the address database so the CPU can control when and where addresses are removed.

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

2.4.7

802.1X Source MAC Address Checking

The device supports 802.1X source MAC address authentication using CPU Directed Address Learning ([Section 2.4.6](#)) along with the DropOnLock Ingress policy ([Section 3.1.2](#)). CPU Directed Address Learning is required for 802.1X³ so that the requesting MAC address can be authorized by the authorization server. The DropOnLock policy causes all frames from unauthorized MAC addresses to be discarded.

A side effect of authorization is that a CPU might become saturated from constant SA Miss Violations from a source that it has denied. This is prevented in the device by masking denied MAC addresses. An address can be masked by loading it into the address database with a Destination Port Vector (DPV) of all zeros⁴. The address appears in the database so it no longer causes a 'Miss' Violation. Any port trying to send a frame to this unauthorized address is discarded (since its DPV is all zeros) and any 802.1X port trying to use this unauthorized address has its frames discarded too (since the port's SA bit is not set in the ATU entry). But now the CPU will get SA Member Violation interrupts every time the unauthorized address is attempted to be used as an SA. These interrupts can be masked too by setting the IgnoreWrongData bit (Port Association Vector – Port offset 0x0B) on the 802.1X ports.

The CPU can mask unauthorized MAC addresses by loading them into the address database as static or non-static entries. If they are loaded non-static, the interrupts are masked until the entry ages out of the database or until the CPU purges the entry (do not enable the port's RefreshLocked feature, Port offset 0x0B, in this case as the refresh will 'authorize' the MAC address by updating the entry's DPV). This approach minimizes the number of interrupts the CPU needs to service while

1. The Age Time (Global 1, offset 0x0A) determines the age time as well. See [Section 2.4.5](#).
2. The association may be direct from the entry's DPV or indirect from the entry's Trunk ID mapped through the Trunk Mapping Table (Global 2 offset 0x08).
3. An 802.1X port needs to have its LockedPort bit set to one and its SAFFiltering bits set to 0x1 (see the Port Association Vector, Port offset 0x0B, and the Port Control register, Port offset 0x04).
4. The all zero DPV cannot be used to mask these interrupts if the enhanced 802.1X Drop to CPU mode is being used to trap semi-authorized frames to the CPU ([Section 3.1.2.2](#)).

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

If a port is saturating a CPU by constantly using a new SA, the masking of unauthorized addresses is not applicable. Instead all SA Miss Violations can be masked on a port by disabling learning on the port (i.e., by setting the port's Port Association Vector bits to all zero (Port offset 0x0B). This configures the ingress port in a form of a Secure Port ([Section 3.1.3](#)) and will work as long as no new SA needs to be authorized on this port.

2.4.8

Multiple Address Database Support (FID)

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

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

- The FID associated with the frame's VID¹ in the VLAN Translation Unit (VTU, [Section 7.2](#)). This requires the frame's VID to be valid in the VTU. All frames are assigned a VID, even untagged frames. This is true if 802.1Q is enabled on the port or not.
- The FID associated with the ingress port (FID[11:4] in Port Control 1 register, Port offset 0x05 and FID[3:0] in Port Based VLAN Map register, Port offset 0x06).

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

A frame's FID is generally determined by the frames VID (via the VTU, [Section 7.2](#)). Multiple VID's can be mapped to the same FID allowing for shared VLAN address databases.

1. The frame's VID can be overridden. See [Section 3.2.2.7](#).



Link Street®

88E6321/88E6320 Functional Specification

3

Normal Network Ports

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

3.1

Ingress Policy

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

- Non-management frame types can be blocked from entering the switch to support Spanning Tree Protocol ([Section 3.1.1](#) for classic 802.1D and [Section 3.2.3](#) for 802.1s).
- The frame's source MAC address can be authenticated and the frame potentially discarded or mapped to the CPU for 802.1X support ([Section 2.4.7](#) and [Section 3.1.2](#)).
- Layer 2 Policy actions (mirror, trap or discard) can be performed based on the frame's DA, SA, VID, Ether type, and/or if the frame is a UDP broadcast and/or a DHCP Option 82.
- Port based VLANs and/or 802.1Q VLANs are used to prevent a frame from going out of certain ports, and switch management Port States, 802.1s (per VLAN spanning tree) or 802.1Q are used to prevent the frame from entering the switch at all ([Section 3.2](#)).
- All IEEE 802.1Q Tagged frames can be discarded ([Section 3.2.2.4](#)) or all Untagged frames can be discarded ([Section 3.2.2.3](#)).
- Port based Ingress Rate Limiting (PIRL) is supported with five independent counters/resources per port ([Section 3.5](#)).
- Any Layer 2 Policy mirror and any Ingress Monitor Source mirror ([Section 6](#))¹ can be statistically sampled using PIRL by mirroring only 1 of every 'n' frames.

3.1.1

Port States Filtering for 802.1D Spanning Tree

The device supports four 802.1D Port States per port shown in [Table 5](#) (802.1s per VLAN Port States are also supported - see [Section 3.2.3](#)). The 802.1D Port States are used by the Queue Controller (see [Section 3.6](#)) in the device to adjust buffer allocation. They are used by the Ingress Policy blocks to control which frame types are allowed to enter and leave the switch so that Spanning Tree or bridge loop detection software can be supported. The PortState bits in the Port Control register (Port offset 0x04) determine each port's Port State (assuming 802.1s per VLAN Port States are not being used) and the bits can be modified at any time without causing any errors on transmitted frames.

[Table 5](#) lists the Port States and describes them. Two of the Port States require the detection of management (MGMT) frames. MGMT frames are defined in [Section 6.2](#). Their primary purpose is to support the Spanning Tree Protocol (see [Section 6.2](#)) so these frames have the ability to tunnel through blocked ports. SA learning is not performed on MGMT frames. MGMT frames also ignore VLAN rules on ingress and egress (802.1Q and Port Based), IGMP snooping and Rate Limiting (if MGMT frames are selected for non-rate limiting in PIRL – [Section 3.5](#)). This means they always go to the port indicated by the Destination Port Vector (DPV) assigned to the frame's DA in the address database or to the device's CPUDest port (Global 1 offset 0x1A) depending upon what MGMT detection mode was used ([Section 6.2](#)). MGMT frames are typically used for 802.1D Spanning Tree

1. ARP Mirrors ([Section 3.3.3](#)) cannot be statically sampled.

Bridge Protocol Data Units (BPDUs), but any multicast or any unicast address can be used supporting new or proprietary protocols.

Table 5: Port State Options

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

- Frames can be filtered for many reasons including Layer 2 Policy (Section 3.1.3), Port States (Section 3.1.1 and Section 3.2.3), 802.1X MAC Authentication (Section 3.1.2.1), 802.1Q Violations (Section 3.2.2), reverse 802.1X MAC Authentication (Section 3.1.2.3) or its Tagging did not match what was expected (Section 3.2.2.3 and Section 3.2.2.4).

The default Port State for all the ports in the switch can be either Disabled or Forwarding depending upon the value of the NO_CPU pin (see 88E6321/88E6320 Datasheet for details). The ports come up in the Forwarding Port State unless the NO_CPU is configured for CPU attached mode. This allows the CPU to fully boot before it brings up the ports and starts accepting frames including running bridge loop or spanning tree detection or routing software.



Note

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

3.1.2

Source Address Filtering

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



3.1.2.1

802.1X MAC Address Authentication (Drop on Lock)

All non-MGMT¹ (non-management) frames received on a port with an unauthorized Source MAC Address are discarded if the port's SA Filtering bits are set to 0x1, Drop on Lock mode (Port offset 0x04). In this mode, only frames with an authorized SA (or MGMT frames) are allowed into the switch to be processed further. An SA is considered authorized if it is present in the address database and its contents are associated to the source port where the frame entered the switch². The policy of how these addresses are entered into that address database is controlled by the ATU ([Section 7.3](#)). It is recommended that CPU Directed Address Learning ([Section 2.4.6](#)) be used on ports supporting MAC based 802.1X authentication (i.e., ports in DropOnLock mode). Before the CPU authenticates an address it may want to inspect the contents of some frames. This can be done using Source MAC Frame Trapping ([Section 3.1.2.2](#)). Excessive interrupts from denied Source Addresses can be masked in this mode (see [Section 2.4.7](#)).

3.1.2.2

Source MAC Frame Trapping (Drop to CPU)

Source MAC Frame Trapping works with 802.1X MAC Address Authentication ([Section 3.1.2.1](#)) and it is enabled on a port if the port's SA Filtering bits are set to 0x3, Drop to CPU mode (Port offset 0x04). The purpose of this mode is to get the CPU more data before it authenticates the Source MAC. In standard 802.1X the CPU has to authenticate the SA based on the SA only. This mode allows the trapping of frames from the requesting SA to the CPU for further inspection. Now the CPU can look at the contents of entire frames to help make the authentication decision.

Drop to CPU mode works identically to the Drop on Lock mode ([Section 3.1.2.1](#)) with one difference. It adds the concept of a semi-authorized MAC address where frames with a semi-authorized SA are mapped to the CPU's port (based on the value of CPUDest, Global 1, offset 0x1A). This means that all non-MGMT³ (non-management) frames received on a port with an unauthorized Source MAC Address are discarded. In this mode, only frames with an authorized or semi-authorized SA (or MGMT frames) are allowed into the switch to be processed further. An SA is considered authorized if it is present in the address database and its contents are associated to the source port where the frame entered the switch⁴. An SA is considered semi-authorized if it is present in the address database and its Destination Port Vector (DPV and 'T' bit) are all zeros. This all zeros value ensures that no other port can transmit frames to this SA (as frames using this MAC address as a DA will be discarded due to the all zero DPV) while allowing frames with that SA to be sent to the CPU for further inspection.

The policy of how these addresses are entered into that address database is controlled by the ATU ([Section 7.3](#)). It is recommended that CPU Directed Address Learning ([Section 2.4.6](#)) be used on ports supporting MAC based 802.1X authentication (i.e., ports in DropOnLock mode).

- Drop to CPU frames egress the CPU's port as a non-MGMT frame and they are filtered based upon the egress port's Port State (Port offset 0x04). If the CPUDest port on the device is using DSA tags ([Section 6](#)) the Drop to CPU frames will egress as DSA Forward frames. In multi-chip or stacking systems, all switch devices in the path from the original ingress port to the actual CPU must have the semi-authenticated MAC addresses entered into their address database or the CPU will not get these trapped frames. These MAC addresses can be entered with SA Priority Override if needed ([Section 3.4.6](#)).
- Excessive interrupts from denied SA's cannot be supported in this mode (see [Section 2.4.7](#)). If this is needed use 802.1X Drop on Lock ([Section 3.1.2.1](#)) and then frames can be trapped to the CPU using Layer 2 Policy.

1. See [Section 6.2](#) on how to detect a MGMT frame.

2. If the port is a non-Trunk port then the ATU entry must not be a Trunk entry and it must have the port's bit set in its DPV. If the port is a Trunk port then the ATU entry must be a Trunk entry matching the Trunk ID of the ingressing port. See [Section 6.10](#).

3. See [Section 6.2](#) on how to detect a MGMT frame.

4. If the port is a non-Trunk port then the ATU entry must not be a Trunk entry and it must have the port's bit set in its DPV. If the port is a Trunk port then the ATU entry must be a Trunk entry matching the Trunk ID of the ingressing port. See [Section 6.10](#).

3.1.2.3

Reverse 802.1X MAC Address Authentication (Drop on Unlock)

Reverse 802.1X MAC address authentication accepts frames from all Source MAC Addresses except for those MAC Addresses that are specifically identified to be denied. When this feature is used together with Address Learn Limiting ([Section 2.4.4](#)) or CPU Directed Learning ([Section 2.4.6](#)), they can be used to prevent Denial of Service (DoS) attacks.

All non-MGMT¹ (non-management) frames received on a port with a semi-authorized Source MAC Address are discarded if the port's SA Filtering bits are set to 0x2, Drop on UnLock mode (Port offset 0x04). In this mode, frames with an unknown or known SA (or MGMT frames) are allowed into the switch to be processed further. An SA is considered known if it is present in the address database and its contents are non-zero². The known addresses can enter the address database through auto learning ([Section 2.4.3](#)) or by CPU directed learning ([Section 2.4.6](#)). An SA is considered semi-authorized if it is present in the address database and its Destination Port Vector (DPV and 'T' bit) are all zeros. This all zeros value ensures that no other port can transmit frames to this SA (as frames using this MAC address as a DA will be discarded due to the all zero DPV) while at the same time ensure all frames with that SA are discarded as well.

This mechanism can be used to shut down the source of a Denial of Service attack by discarding all frames coming from (and to) the identified Source Address of the attacker. If the attacker uses more than one Source Address, these too can be shut down in the same way. The number of 'denied' MAC addresses can be easily limited by combining this feature with Address Learn Limiting ([Section 2.4.4](#)) or CPU Directed Learning ([Section 2.4.6](#)) that performs the same function as Address Learn Limiting.

3.1.2.4

Static or Dynamic Addresses

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

- The 802.1X Drop on Lock mode cannot accept frames unless the frame's SA is in the address database associated with the ingressing port. At first look these should be loaded as static entries. But since there could be a large number of authorized MAC addresses being used at one time, it is probably better to load the authorized entries as dynamic. These addresses can be prevented from aging out if the port's HoldAt1 bit is set (Port offset 0x0B) and they can be self refreshed after being authenticated by enabling the port's Refresh Locked bit (Port offset 0x08B). If an address gets bumped out by a more recently used one, the CPU can quickly reload the already authenticated address that is needed if the CPU keeps a local cache of approved addresses. Alternatively, the CPU can perform the refreshes by configuring the device such that it gets informed about addresses that are being used whose age (or EntryState) is less than 0x4 by setting the port's IntOnAgeOut bit (Port offset 0x0B). This approach prevents the CPU from running into an ATU Full condition where a MAC address cannot be loaded (see [Section 7.3.6](#)).
- The 802.1X Drop to CPU mode should load authorized addresses as dynamic (see above), but the semi-authorized addresses must be loaded as static or only the 1st frame will be trapped to the CPU and all others will be accepted as being authorized without CPU intervention. It is assumed that the CPU will only need to look at the contents of a few frames and then it will either fully authorize the MAC or fully de-authorize it.
- The Reverse 802.1X Drop on UnLock mode allows automatic address learning with very few addresses that may need to be loaded to prevent a Dos attack. Since the number of these semi-authorized addresses is very low, the CPU can load these addresses as static. But the CPU will need to keep track of these addresses and unload them after some period of time so the port can start accepting frames again. Alternatively, the CPU can load these addresses as dynamic disallowing frames for a period of the AgeTlme (typically 5 min. – Global 1 offset 0x0A) and re-loading them as needed.

1. See [Section 6.2](#) on how to detect a MGMT frame.

2. The contents of an ATU entry are considered zero if the entry's 'T' bit and the entry's DPV bits are zero ([Section 7.3.1](#)).

3.1.3

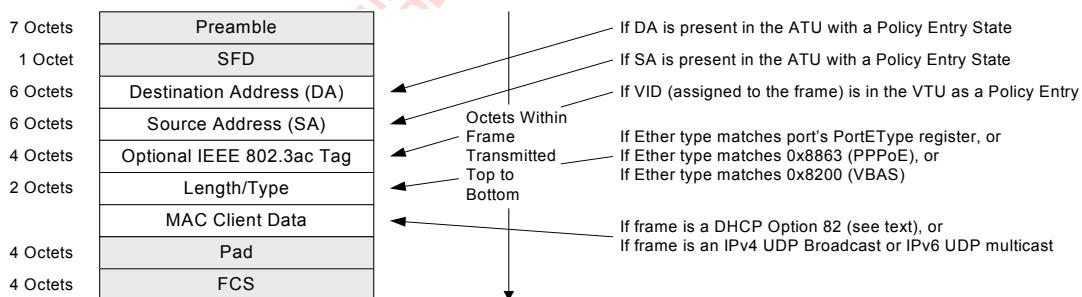
Layer 2 Policy Control Lists

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

- Normal frame switching (i.e., do nothing special)
- Policy Mirror (copy) the frame to the MirrorDest port¹ (Global 1 offset 0x1A)
- Policy Trap (re-direct) the frame to the CPUDest port (Global 1 offset 0x1A)
- Policy Discard (filter) the frame

The fields in the frames where the above policy is selectable on a per port basis are shown in Figure 2.

Figure 2: Fields of the Frame Examined for Layer 2 PCL



Ingressing Frame

Layer 2 PCLs

Each port supports separate actions for each of the eight possible policy items (Port offset 0x0E). Therefore, a frame could get more than one policy action applied to it. A single frame can be both Policy Trapped (to CPUDest) and Policy Mirrored (to MirrorDest), but if any policy action is Policy Discard, the frame is discarded without being mapped anywhere else. Likewise, if a frame is discarded for other switch policy reasons (like VLAN membership, [Section 3.2.2](#), or because it is tagged or untagged, [Section 3.2.2.4](#) and [Section 3.2.2.3](#)) the frame will not be Policy Mirrored or Policy Trapped either.

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

3.1.3.1

DA, SA, and VID Policy

As each frame enters the switch, both its DA and SA are looked up into the address database. If the DA is found in the ATU with a Policy Entry State ([Section 7.3.1](#)) then Layer 2 Policy will occur on that frame on the ports where DA Layer 2 PCLs are enabled (Port offset 0x0E). If the SA is found in the ATU with a Policy Entry State then Layer 2 Policy will occur on that frame on the ports where SA Layer 2 PCLs are enabled.

As each frame enters the switch, a VID is extracted or assigned to the frame ([Section 3.2.2.7](#)) and then that VID is looked up in the VLAN database. If the VID is found in the VTU with its Policy bit set

1. Any mirror in the 88E6321/88E6320 Functional Specification device can be sampled. See [Section 3.5](#).

to a one ([Section 7.2](#)) then Layer 2 Policy will occur on that frame on the port's where VTU Layer 2 PCLs are enabled (Port offset 0x0E).



Note

- Each MAC and VID entry is globally flagged as a Policy entry or not (see [Section 7.3.1](#) for MAC and [Section 3.2.2.1](#) for VID). So the same MAC address can become a Policy Discard for Ports 5 to 0, a Policy Trap for Port 6, and do normal switching for Port 7 all at the same time. But once the ports are configured this way, all MAC addresses with a Policy Entry State in the ATU will work the same way.
- DA and SA Policy entries in the ATU are static entries so auto leaning cannot change their values. The CPU will need to manually purge these entries once Layer 2 Policy is no longer needed on these addresses.

3.1.3.2 Ether Type Policy

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



Note

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

3.1.3.3 DHCP Option 82

DHCP Option 82 is a special policy function that goes beyond the layer 2 fields of the frame. If the frame entering a port matches either of the frame formats shown in [Figure 3](#) or [Figure 4](#), then Layer 2 Policy will occur on that frame if the port's Opt82 Layer 2 PCL is enabled (Port offset 0x0E). DHCP Option 82 is supported for both IPv4 and IPv6. The portions of the frame that must match are those fields in the figures that are in a white background. Those fields in a grey background are not examined for this feature.

3.1.3.4 UDP Broadcasts

UDP Broadcast is another special policy function that goes beyond the layer 2 fields of the frame. If the frame entering a port matches either of the frame formats shown in [Figure 3](#) or [Figure 4](#), then Layer 2 Policy will occur on that frame if the port's UDP Layer 2 PCL is enabled (Port offset 0x0E). UDP Broadcast is supported for both IPv4 and IPv6 (although in IPv6 the frame only has to be a multicast). The portions of the frame that must match are those fields in the figures that are in a white background. Those fields in a grey background are not examined for this feature.

Figure 3: IPv4 DHCP Option 82 Frame Format

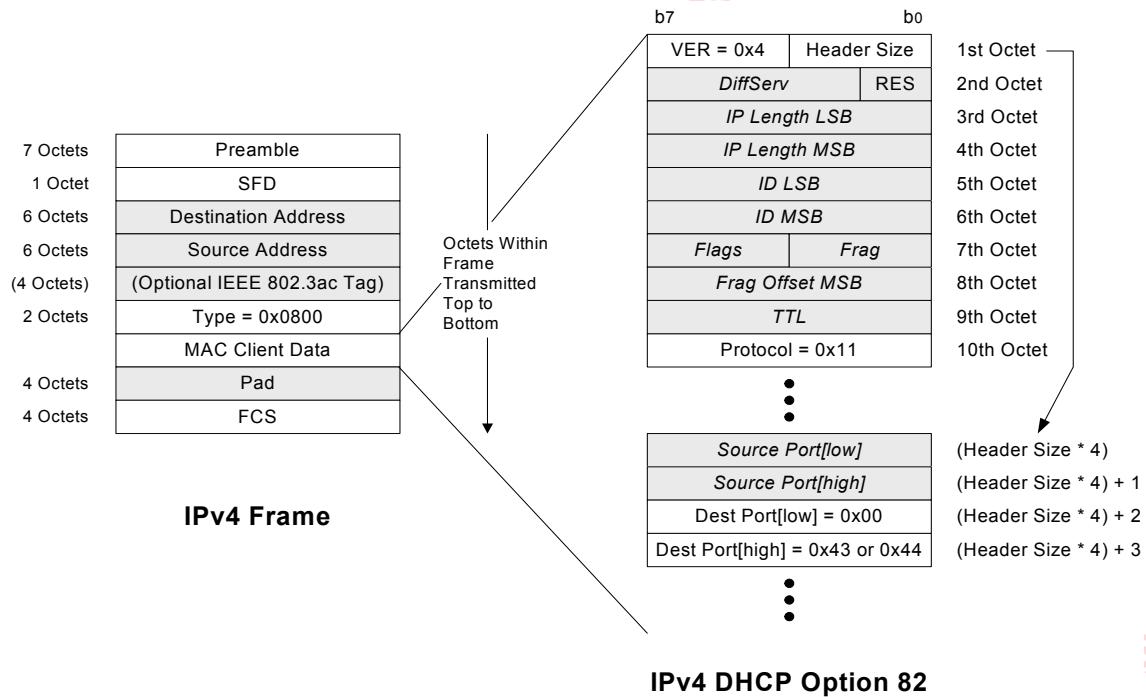
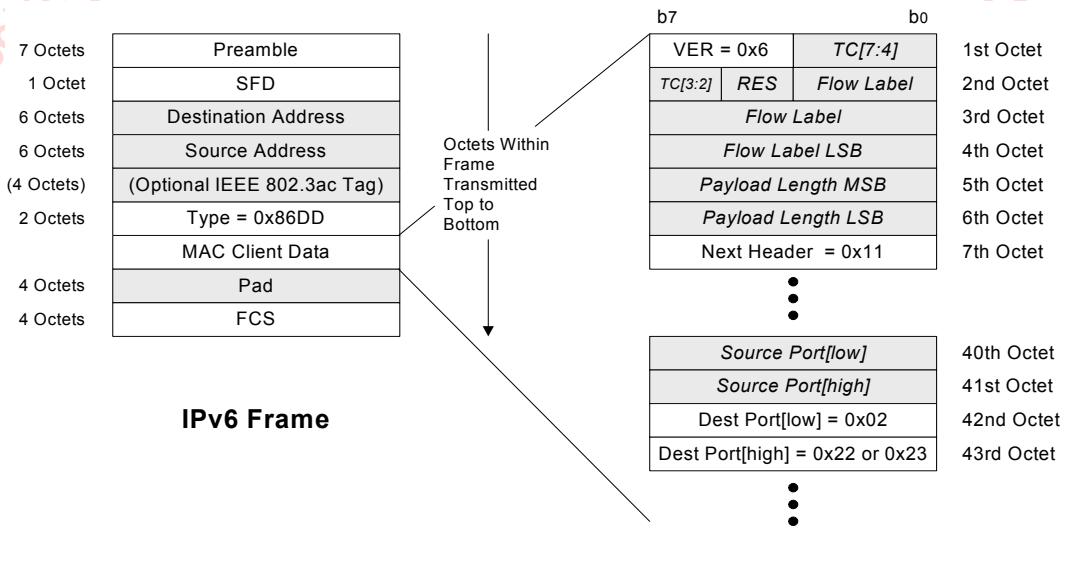


Figure 4: IPv6 DHCP Option 82 Frame Format



3.2 VLANS

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

3.2.1 Port Based VLANs

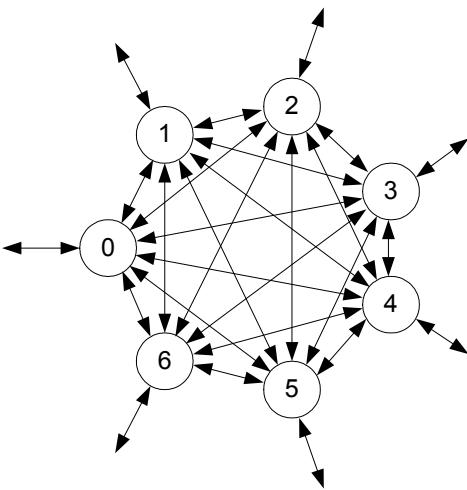
The device supports port based VLANs both in-chip (within a single device) and cross-chip (cascading across multiple devices – [Section 5.5.3](#)).

3.2.1.1 In-chip Port Based VLANs

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

Each Ingress port contains a register that restricts the output (or egress) ports to which it is allowed to send frames. This register is called the VLANTable register (Port offset 0x06). If bit 0 of a port's VLANTable register is set to a one, that port is allowed to send frames to Port 0. If bit 1 of a port's registers is set to a one, that port is allowed to send frames to Port 1. Bit 2 for Port 2, etc. At reset the VLANTable register for each port is set to a value of all one's, except for each port's own bit, which is cleared to a zero (this prevents frames from going back out of the port they came in on¹). This default VLAN configuration allows all the ports to send frames to all the other ports as shown in [Figure 5](#).

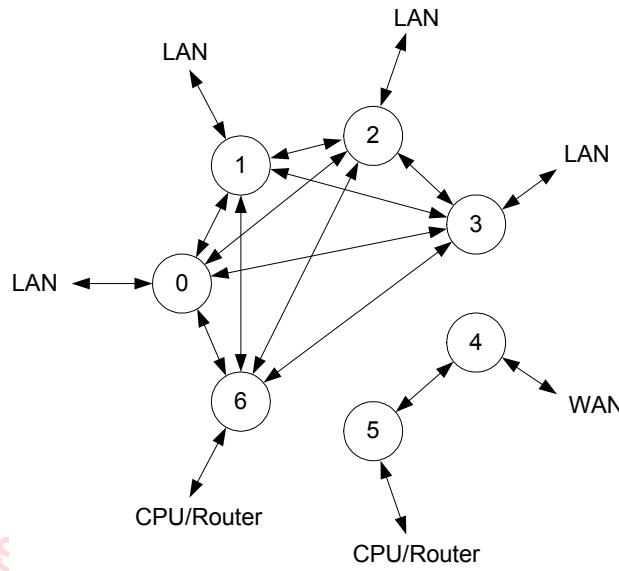
Figure 5: Switch Operation with Port VLANs Disabled



3.2.1.2 Port Based VLAN Router Examples

One of the main applications for port based VLAN support in the device is to isolate a port or ports for firewall router applications. [Figure 6](#) shows a typical VLAN configuration for a firewall router. Port 4 is used as the WAN port. The data coming in from this WAN port must not go out to any of the LAN ports – but it must be able to go to the router CPU. All the LAN ports are able to send frames directly to each other without the need of CPU intervention – but they cannot send frames directly to the WAN port. The CPU is able to send frames to all of the ports so that routing can be accomplished. The use of the Marvell® Header² ([Section 6.7](#)), enables a CPU to define dynamically which port or ports a particular frame is allowed to reach for purposes of WAN and LAN isolation on multicast traffic generated by the CPU³.

1. The device allows a port's own bit in its VLANTable to be set to a one – see [Section 3.3.1](#)
2. The Marvell Header is designed to be used on a CPU port only.
3. The Marvell Header can be used to isolate ports on multicast traffic in single chip implementations only – i.e., it is for routers.

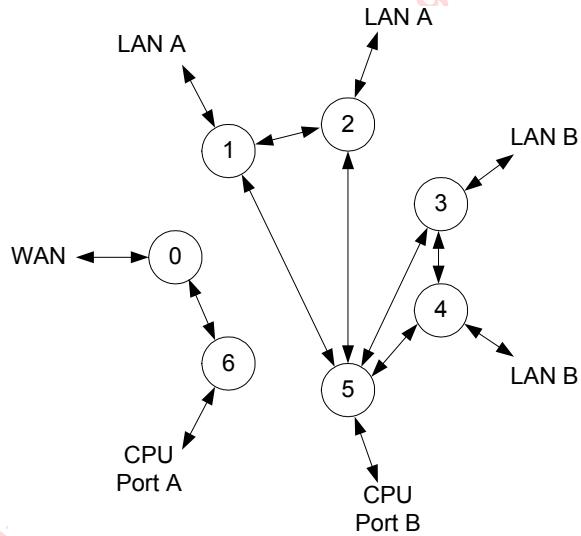
Figure 6: Switch Operation with a Typical Router VLAN Configuration

The example VLAN configuration shown in [Figure 6](#) is achieved by setting the port's VLANTable registers as shown in [Table 6](#):

Table 6: VLANTable Settings for Figure 6

Port #	Port Type	VLANTable Setting
0	LAN	0x4E
1	LAN	0x4D
2	LAN	0x4B
3	LAN	0x47
4	WAN	0x20
5	CPU	0x10
6	CPU	0x0F

To show the flexibility of the device VLAN configuration options, [Figure 7](#) shows another example. In this case, the switch is divided into three independent VLANs connected to a common router.

Figure 7: Switch Operation with another Example VLAN Configuration

The example VLAN configuration shown in [Figure 7](#) is accomplished by setting the port's VLANTable registers as shown in [Table 7](#):

Table 7: VLANTable Settings for Figure 7

Port #	Port Type	VLANTable Setting
0	WAN	0x40
1	LAN A	0x24
2	LAN A	0x22
3	LAN B	0x30
4	LAN B	0x28
5	CPU B	0x1E
6	CPU A	0x01

3.2.2

802.1Q VLANs

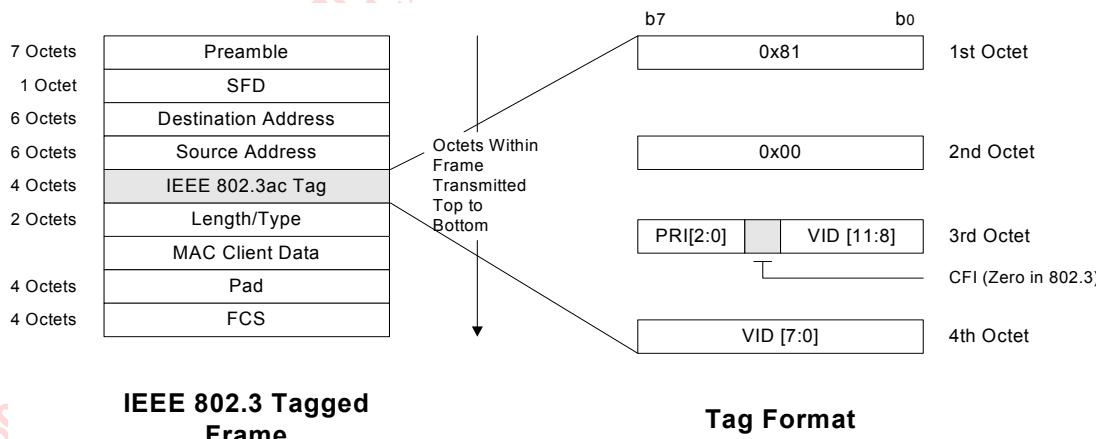
The device supports 802.1Q with the full set of 4,096 different VID (VLAN identifiers). Some or all of the VIDs can be used (i.e., software only needs to initialize the VIDs that are being used). Since the device may be programmed with only a subset of the possible VIDs, and security requirements vary, the device supports 802.1Q in three different modes. The device's port-based VLAN feature (both in-chip and cross-chip, [Section 3.2.1](#) and [Section 5.5.3](#)) is in effect for all 802.1Q modes described below. 802.1Q VLANs are supported cross-chip as well ([Section 5.5.2](#)).

3.2.2.1

IEEE Tagging VID Handling

VLAN Identifiers (VIDs) are contained in IEEE tagged frames. All frames ingressing the device are assigned a VID, even the untagged frames. The format of an IEEE tagged frame is shown in [Figure 8](#). The discussion below is relevant only if the port's Frame Mode is Normal Network (Port offset 0x04).

Figure 8: IEEE Tag Frame Format



3.2.2.2

Determining if a Frame is Tagged

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

3.2.2.3

Discarding Untagged Frames

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

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

3.2.2.4

Discarding Tagged Frames

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

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

3.2.2.5 VID Extraction

If any of the three supported 802.1Q modes are enabled on this port (Section 3.2.2.8) the VID read from tagged frames is assigned to the frame (unless overridden – see Section 3.2.2.6). If the frame's VID = 0x000 the port's DefaultVID (Port offset 0x07) is assigned to the frame instead. If these physically tagged frames egress a port Tagged their VID bits will be overwritten with the assigned value.


Note

If 802.1Q is disabled on this port the VID bits from tagged frames are ignored, and physically tagged frames are considered physically untagged for egress tag processing (i.e., transmit Tagged will add a tag and transmit UnTagged or Unmodified will transmit the frame unmodified - Section 3.8.4). These frames are assigned the ingress port's DefaultVID (Port offset 0x07) which will be written to the frame's new (added) VID bits if the frame egresses a port Tagged.

3.2.2.6 Security Override of a Frame's VID

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

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

3.2.2.7 VID Assigned to the Frame

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

If a frame entering the switch is untagged, it is assigned the port's DefaultVID during Ingress (Port offset 0x07). If a frame is tagged its VID is generally used as the frame's VID unless the frame's VID is 0x000 or if the port's ForceDefaultVID is set. A summary of how a VID is assigned to each frame is shown in Table 8.

Table 8: Example VID Assignment Summary

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



Link Street®

88E6321/88E6320 Functional Specification

3.2.2.8

Security & Port Mapping

The 802.1Q Security features of the device supports the discarding of ingressing frames that don't meet the security requirements and ensuring that those frames that do meet the requirements are sent to the allowed ports only. Three levels of security are supported and they can be set differently on each port. The security options are processed using the VID assigned to the frame ([Section 3.2.2.7](#)) as follows:

- Secure – The VID must be contained in the VTU and the Ingress port must be a member of the VLAN else the frame is discarded. The frame is allowed to exit only those ports that are both:
 - Members of the frame's VLANand
 - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see [Section 3.2.1](#))
- Check – The VID must be contained in the VTU or the frame is discarded (the frame will not be discarded if the Ingress port is not a member of the VLAN). The frame is allowed to exit only those ports that are both:
 - Members of the frame's VLANand
 - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see [Section 3.2.1](#))
- Fallback – Frames are not discarded if their VID is not contained in the VTU.
 - If the frame's VID is contained in the VTU, the frame is allowed to exit only those ports that are both:
 - Members of the frame's VLANand
 - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see [Section 3.2.1](#))
 - If the frame's VID is not contained in the VTU, the frame is allowed to exit only those ports that are:
 - Included in the source port's port-based VLAN (both In-Chip and cross-chip, see [Section 3.2.1](#))
- 802.1Q Disabled – Frames are not discarded if their VID is not contained in the VTU. The frame is allowed to exit only those ports that are:
 - Included in the source port's VLAN (both In-Chip and cross-chip, see [Section 3.2.1](#))



See 802.1Q Disable Mode Note in [Section 3.2.2.5](#).

Note

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

3.2.2.9

Security Violations

If 802.1Q is enabled on a port, security violations are captured and an interrupt can be generated to the CPU (if unmasked by the VTUProbIntEn bit (Switch Global Control, global offset 0x04)). This is true regardless of the 802.1Q mode (VTUProb interrupts will not occur from a port if the port's 802.1QMode is 802.1Q Disabled - Port offset 0x08)). The interrupts (up to one at a time) are

captured by the VLAN Translation Unit (see VTU Operation register, Global 1 offset 0x05). Two kinds of security violations are captured. A MissViolation occurs if a frame's VID is not contained in the VTU. A MemberViolation occurs if a frame's VID is in the VTU but the source port of the frame is not a member of the frame's VLAN. The security violation captures the offending source port (SPID) and VID that caused the violation. This data is accessed by executing a Get/Clear Violation Data operation in the VTU.

3.2.2.10 Security Override of a Frame's VID

A Tagged frame's VID can be forced to the port's DefaultVID (see [Section 3.2.2.6](#)).

3.2.3 802.1s Per VLAN Spanning Tree

The device supports per VLAN Port States for up to 64 802.1s per VLAN Spanning Tree instances. Each VID entry in the VTU ([Section 7.2](#)) has a 6-bit SID value associated with it that is used to access 1 out of 64 possible 802.1s spanning tree instances from the STU ([Section 7.2.6](#)). Each STU entry contains two bits of per port 802.1s Port State information as shown in [Table 9](#). Using the indirect STU approach allows extremely fast per VLAN spanning tree updates as only one STU entry needs to be modified for all the VIDs using the same spanning tree instance.

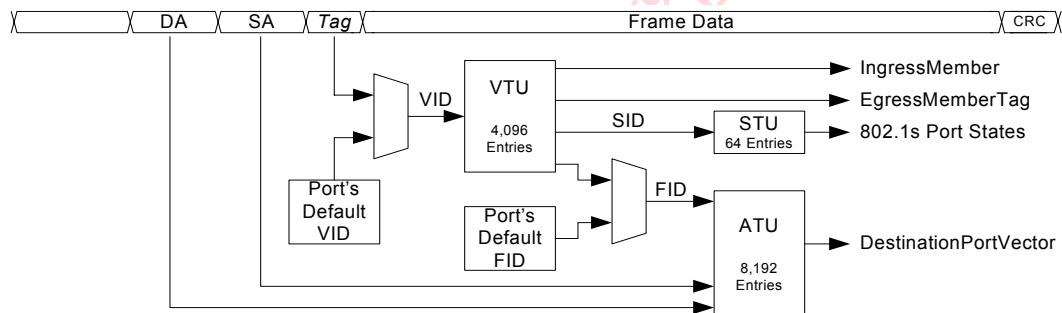
The relationship between the VTU, STU and ATU is shown in [Figure 9](#).

Table 9: 802.1s Port State Options

Port State	Description
802.1s Disabled	The port's PortState bits in the Port Control register (Port offset 0x04) are used for this port for frames with a VID that is associated with this SID. See Section 3.1.1 .
Blocking/Listening	Only MGMT frames are allowed to enter (ingress) or leave (egress) this port for frames with a VID that is associated with this SID. All other frame types are discarded. Learning is disabled on Blocked ports.
Learning	Only MGMT frames are allowed to enter (ingress) or leave (egress) this port for frames with a VID that is associated with this SID. All other frame types are discarded but learning takes place on all good non-MGMT frames that are not discarded owing to being filtered ¹ .
Forwarding	Normal operation. All frames are allowed to enter (ingress) and leave (egress) this port for frames with a VID that is associated with this SID. Learning takes place on all good non-MGMT frames that are not discarded owing to being filtered.

1. Frames can be filtered for many reasons including Layer 2 Policy ([Section 3.1.3](#)), Port States ([Section 3.1.1](#) and [Section 3.2.3](#)), 802.1X MAC Authentication ([Section 3.1.2.1](#)), 802.1Q Violations ([Section 3.2.2](#)), reverse 802.1X MAC Authentication ([Section 3.1.2.3](#)) or its Tagging did not match what was expected ([Section 3.2.2.3](#) and [Section 3.2.2.4](#)).

Figure 9: Relationship between VTU, STU, and ATU





Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

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

If 802.1s is not being used, all STU entries must use the 802.1s Disabled setting for all ports for all SIDs. If 802.1s is being used, all VTU entries must be configured with the required SID and each SID that is being used must define the 802.1s Port State for each port. Any SID entry can contain a mixture of ports using 802.1s along with 802.1D. Those ports using 802.1s need to use a value other than 802.1s Disabled in its SID entry. Those ports using the 802.1s Disabled port state will use the port's Port State setting instead (i.e., 802.1D - [Section 3.1.1](#)).

The 802.1s Port State options take precedence over the port's PortState bits settings ([Section 3.1.1](#)), with the exception of the port's Disabled Port State (set in the port's PortState bits, Port offset 0x04). The port's Disabled Port State prevents all frames from entering and leaving the port so that has precedence over 802.1s Port States.

3.3

Special Frame Handling

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

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

3.3.1

Switching Frames Back to their Source Port

The device supports the ability to send frames back out of the port on which they arrive. While this is not a standard way to handle Ethernet frames, some applications, like 802.3ah OAM loopback ([Section 6.8.2](#)), may require this ability on some ports. This feature can be enabled on a port-by-port basis by setting the port's own bit to a one in its VLANTable register (in the Port Based VLAN Map register, Port offset 0x06). This function is valid if 802.1Q is enabled on the port or not.

3.3.2

Tunneling Frames through VLANs

Normally frames cannot pass between port-based VLANs nor 802.1Q VLANs. The device can be configured to allow some frames to do so. Before a frame can tunnel through a VLAN barrier, its DA address must be loaded as static into the address database (see [Section 7.3.1](#)) and the VLANTunnel bit on the frame's Ingress port must be set to a one (see Port Control register, offset 0x04). When both of these conditions are true, the frame is sent out of the port or ports indicated in the static address's Destination Port Vector (the DPV field for the DA entry in the address database). The VLANTable (for In-chip Port Based VLANs), the cross-chip Port Base VLAN Table and 802.1Q membership data is ignored in this case. This feature is enabled only on those ports that have their VLANTunnel bit set to a one.

3.3.3

ARP Mirroring

The device supports ARP Mirroring on a per port basis. Mirroring is used to copy certain frames to the CPU for processing – tunneling them through VLAN barriers that may be in place to isolate the CPU from unnecessary frames. The required format of these frames is shown in [Figure 10](#). The white portions of the frame in the figures are the only portions of the frame examined for this function.



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

Note

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

Management (MGMT) frames can still be ARP Mirrored (see [Section 6.2](#)). In this case the CPU will get two copies of the frame, one marked as a To_CPU BPDU (MGMT) and another marked as a To_CPU ARP Mirror, assuming the CPU's port is in DSA Tag mode (see [Section 4](#)).

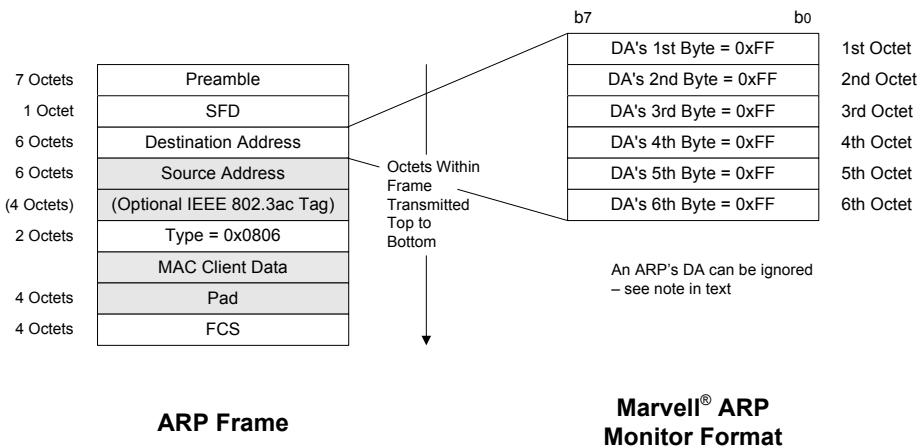
When one of these frames enters a port where ARP Mirroring is enabled (by setting the port's ARP Mirror bit, Port offset 0x08), the frame is copied to the CPU's port as defined by the CPUDest¹

register (Global 1 offset 0x1A). The frame continues to be mapped to all the ports where it would have gone if ARP Mirroring was not enabled on the port. The frame will not get mirrored if it is filtered due to any rules enabled in the Ingress Policy block ([Section 3](#)). The CPU port (as defined by PortDest) does not have to be a member of the frame's VLAN to receive the frame. This is true for Port based VLANs ([Section 3.2.1](#)) and for 802.1Q based VLANs ([Section 3.2.2](#)).


Note

The queue priority (QPri) on ARP frames can be overridden. See [Section 3.4](#).

Figure 10: ARP Mirror Format



3.3.4

IGMP Trapping or Snooping

The device supports IPv4 IGMP snooping and IPv6 MLD snooping on a per port basis. Snooping is used to direct certain frames to the CPU for processing – tunneling them through VLAN barriers that may be in place to isolate the CPU from unnecessary frames. The required formats of these frames are shown in [Figure 11](#) and [Figure 12](#). The white portions of the frame in the figures are the only portions of the frame examined for this function.

Management (MGMT) frames bypass IGMP/MLD snooping (see [Section 6.2](#)).

When one of these frames enters a port where IGMP/MLD snooping is enabled (by setting the port's IGMP/MLD Snoop bit, Port offset 0x04), the frame is sent to the CPU's port as defined by the CPUDest register (Global 1 offset 0x1A) instead of where the frame normally would have been mapped. The frame will not get sent to the CPU if it is filtered due to any rules enabled in the Ingress Policy block ([Section 3](#)). The CPU port (as defined by PortDest) does not have to be a member of the frame's VLAN to receive the frame. This is true for Port based VLANs ([Section 3.2.1](#)) and for 802.1Q based VLANs ([Section 3.2.2](#)).


Note

The queue priority (QPri) on IGMP/MLD snooped frames can be overridden. See [Section 3.4](#).

1. ARP Mirrors always go out the port mapped by CPUDest. They do not go out the MirrorDest port which is reserved for Policy Mirrors ([Section 3.1.3](#)).

Figure 11: IPv4 IGMP Snoop Format

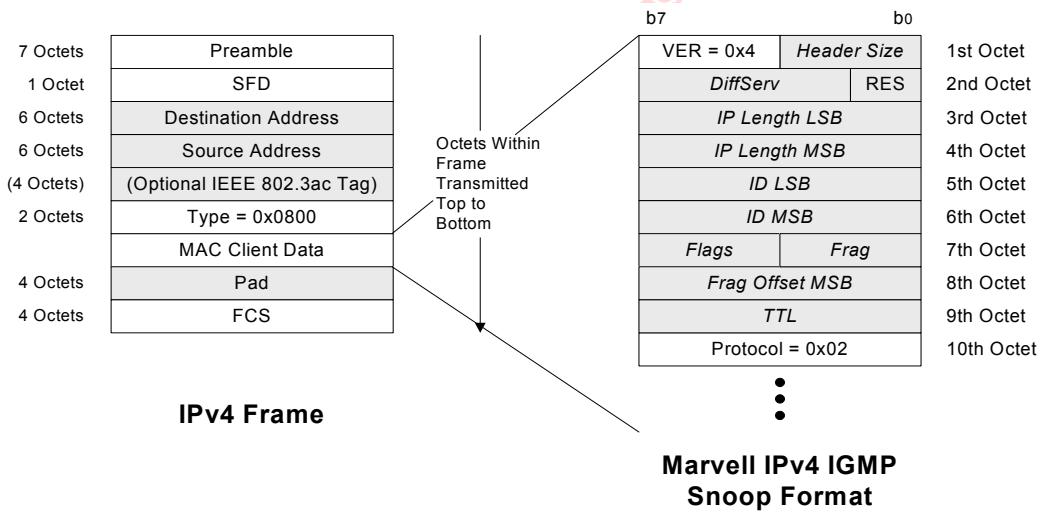
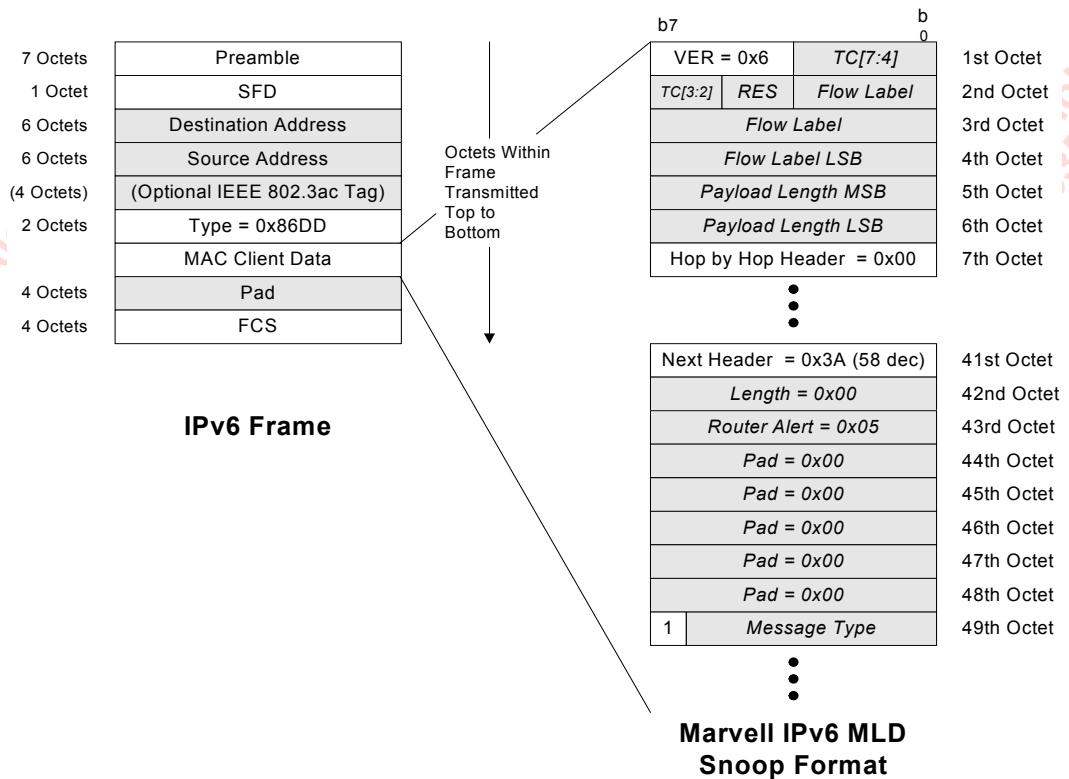


Figure 12: IPv6 MLD Snoop Format



3.4

Quality of Service (QoS) Classification

The Ingress block has the task of determining the priority of each frame to be used for the internal Queue Controller (QPri) as well as the priority assigned to the frame (FPri) if the frame egresses the switch tagged ([Section 3.8.4](#)). The Ingress block does not perform the QoS switching policy, which is the task of the Queue Controller ([Section 3.6](#)). Instead, it has the job of determining the QPri and FPri assigned to each frame for the Queue Controller and Egress block. The priority of a frame is determined by the following process (in this process order, therefore the last process step determines the final priority).

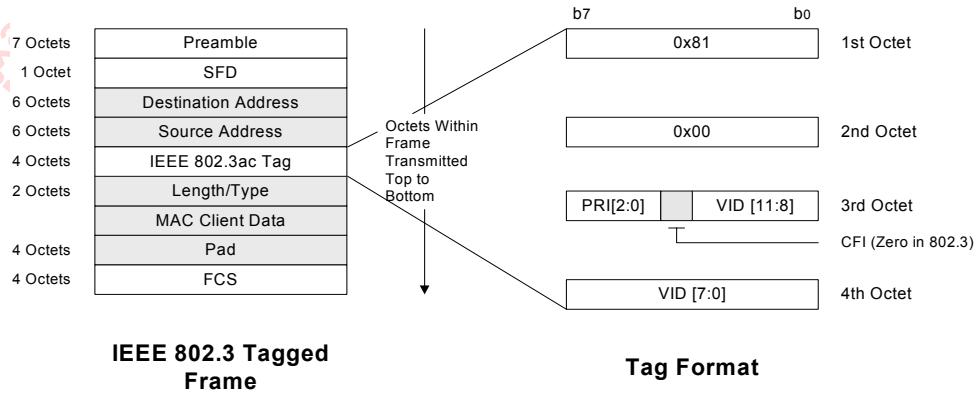
1. IEEE Tagged Frame Priority Extraction ([Section 3.4.1](#))
2. IPv4 and IPv6 Frame Priority Extraction ([Section 3.4.2](#))
3. Default Priority ([Section 3.4.3](#))
4. Initial Priority Override ([Section 3.4.4](#))
5. Frame Type Priority Override ([Section 3.4.5](#))
6. Layer 2 Priority Override ([Section 3.4.6](#))

3.4.1

IEEE Tagged Frame Priority Extraction

All ingressing frames with an 0x8100 ether type right after the frame's Source Address (see [Figure 13](#)) have their IEEE Tagged PRI bits mapped into the port's 8 entry x 3 bit IEEE Priority Remapping table ([Figure 14](#)- port offsets 0x18 and 0x19). The result of this mapping is assigned as the frame's current FPri (the frame's final FPri value is remarked into the frame's IEEE Tagged PRI bits if the frame egresses a port tagged). This assignment occurs if 802.1Q is disabled or enabled on the port and occurs regardless of the frame's VID.

Figure 13: IEEE Tag Format



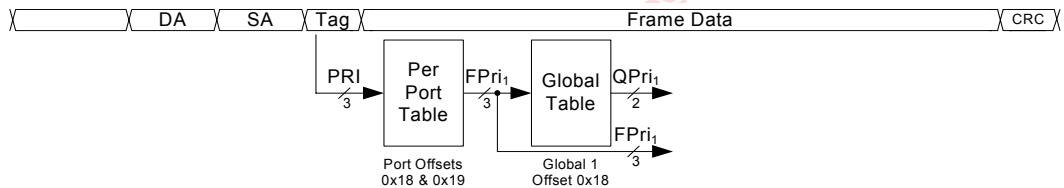
IEEE 802.3 Tagged Frame

Tag Format

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

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

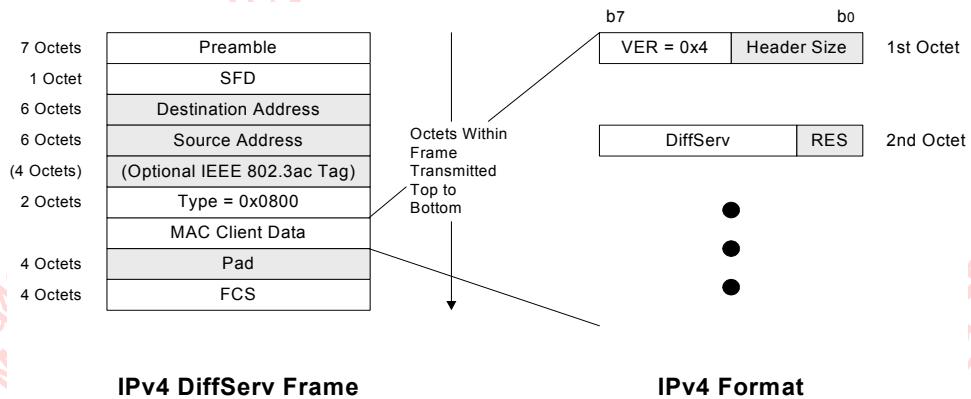
Figure 14: Port's IEEE PRI Mapping



3.4.2 IPv4 and IPv6 Frame Priority Extraction

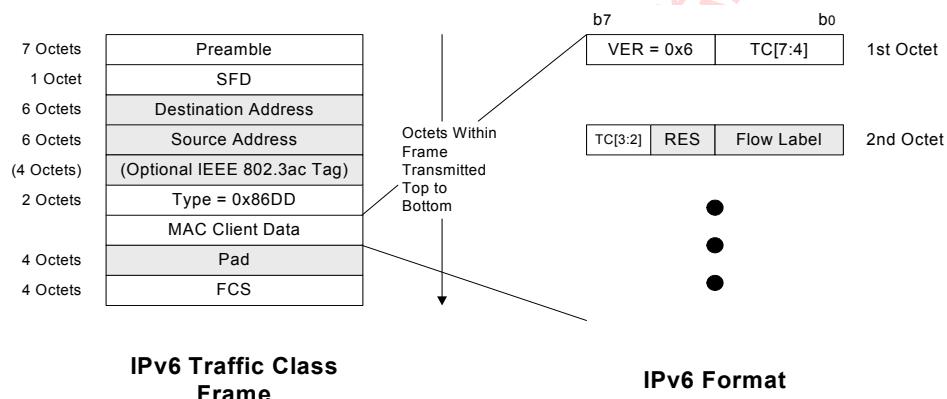
All ingressing frames with an 0x0800 ether type (IPv4) right after the frame's Source Address or right after an optional IEEE Tag (see [Figure 15](#)) have their VER bits checked. If the VER bits = 0x4 then the DiffServ bits are assigned as the frame's IP_PRI bits. These IP_PRI bits can be used to determine the frame's QPri and/or FPri bits (see [Section 3.4.4](#)).

Figure 15: IPv4 Priority Frame Format



All ingressing frames with an 0x86DD ether type (IPv6) right after the frame's Source Address or right after an optional IEEE Tag (see [Figure 16](#)) have their VER bits checked. If the VER bits = 0x6 then the upper 6 bits of the Traffic Class (TC[7:2]) are assigned as the frame's IP_PRI bits. These IP_PRI bits can be used to determine the frame's QPri and/or FPri bits (see [Section 3.4.4](#)).

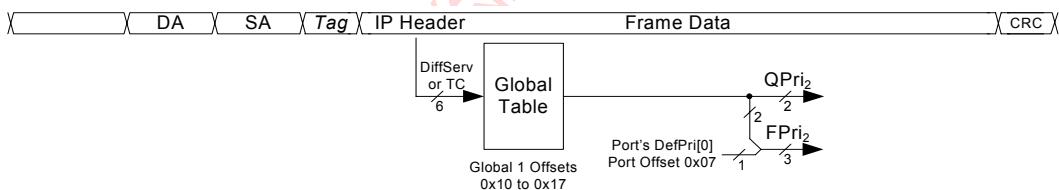
Figure 16: IPv6 Priority Frame Format



In both the IPv4 and IPv6 cases, the frame's IP_PRI bits are mapped into the global IP PRI Mapping Table (Global 1, offsets 0x10 to 0x17) in order to assign a QPri (queue priority) to the frame (Figure 17). The QPri bits are used as the upper two bits of the frame's FPri (frame priority). The least significant bit of the FPri in this case comes from the least significant bit of the port's DefPri (Default Priority, port offset 0x07).

If these frames egress out a port as IEEE tagged, the FPri value will be written to the frame's PRI bits (see Figure 13). This allows the IP priority to determine the tagged priority assigned to the frame.

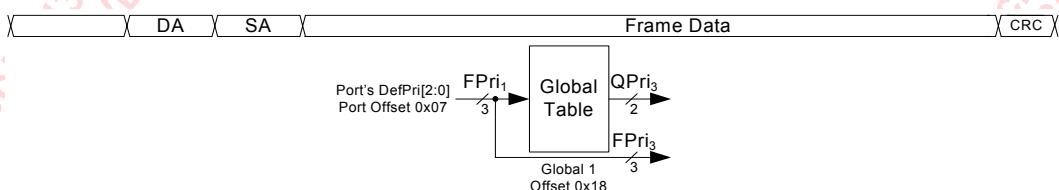
Figure 17: Port's IP Pri Mapping



3.4.3 Default Priority

All ingressing frames are assigned a default FPri (frame priority) and QPri (queue priority). The port's Default Priority (DefPri, port offset 0x07) is assigned as the frame's FPri. The FPri is then mapped into the global IEEE PRI Mapping Table (Global 1, offset 0x18) in order to assign a QPri to the frame (see Figure 18).

Figure 18: Port's Default PRI Mapping



3.4.4 Initial Priority Selection

Every frame entering the switch gets two priority values assigned to it. One priority value is used inside the switch only to determine which output queue the frame is to be mapped into. This is called QPri for queue priority. The other priority value is used outside the switch only to mark the frame's PRI bits if the frame egresses a port tagged (see Figure 13). This is called FPri for frame priority.

The QPri and FPri values are assigned differently to various frame types (see Section 3.4.1 to Section 3.4.3). Some frames are assigned more than one set of QPri and FPri values. For example an IEEE tagged IPv4 frame will get a set of values from the IEEE tag, another set from the IPv4 header and another set from the port. Which set should be used? Some applications require that the port's default priority be used regardless of the contents of the frame. How can this be accomplished? Table 10 shows how the port's InitialPri register bits along with the port's TagIfBoth

register bit (both at Port offset 0x04) can be used to control which QPri and FPri values are assigned to a frame (prior to any priority overrides that may occur – [Section 3.4.5](#) to [Section 3.4.6](#)).

Table 10: Initial QPri and FPri Selection

Selection	InitialPri	Ingressing Frame Type	TagIfBoth	Assigned Pri's	Figure and Meaning
Use port's defaults	0x0	All frame types	Don't Care	QPri = QPri ₃ FPri = FPri ₃	QPri = QPri ₃ /FPri = FPri ₃ See Figure 18 - Use port's defaults
Support Tag priority only	0x1	Untagged frames	Don't Care	QPri = QPri ₃ FPri = FPri ₃	QPri = QPri ₃ /FPri = FPri ₃ See Figure 18 - Use port's defaults
		Tagged frames		QPri = QPri ₁ FPri = FPri ₁	QPri = QPri ₁ /FPri = FPri ₁ See Figure 14 - Use Tag's remapped priority
Support IPv4 or IPv6 priority only	0x2	Neither IPv4 nor IPv6 Frames	Don't Care	QPri = QPri ₃ FPri = FPri ₃	QPri = QPri ₃ /FPri = FPri ₃ See Figure 18 - Use port's defaults
		IPv4 nor IPv6 Frames		QPri = QPri ₂ FPri = FPri ₂	QPri = QPri ₂ /FPri = FPri ₂ See Figure 17 - Use IP priority
Support IPv4, IPv6 and Tag priority	0x3	Neither IPv4 nor IPv6 nor tagged frames	Don't Care	QPri = QPri ₃ FPri = FPri ₃	QPri = QPri ₃ /FPri = FPri ₃ See Figure 18 - Use port's defaults
		Untagged IPv4 or IPv6 frames		QPri = QPri ₂ FPri = FPri ₂	QPri = QPri ₂ /FPri = FPri ₂ See Figure 17 - Use IP priority
		Tagged frames that are not IPv4 nor IPv6		QPri = QPri ₁ FPri = FPri ₁	QPri = QPri ₁ /FPri = FPri ₁ See Figure 14 - Use Tag's remapped priority
		Tagged IPv4 and Tagged IPv6 frames	0x0	QPri = QPri ₂ FPri = FPri ₁ (Special Case)	QPri = QPri ₂ /FPri = FPri ₁ See Figure 17 for IP QPri – Use IP QPri See Figure 14 for Tag FPri – Use Tag's remapped FPri priority
			0x1	QPri = QPri ₁ FPri = FPri ₁	QPri = QPri ₁ /FPri = FPri ₁ See Figure 14 - Use Tag's remapped priority



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

TagIfBoth being 0x0 controls only the selection of the internal QPri on frames that are both tagged and IP (with an InitialPri setting of 0x3). This allows the PRI bits in tagged frames to still come from the port's IEEE Priority Remapping table while the internal QPri comes from the IP portion of the frame. If the PRI bits should be determined based on the IP priority of the frame instead, use an InitialPri setting of 0x2.

The InitialPri bits are enables on each of the priority classifications giving the designer any combination that is needed. For instance, if a port based higher priority port is required for a switch design the priority classification selections can be disabled by setting InitialPri to 0x0 on the port.



This leaves the port's default priority resulting in all Ingressed frames being assigned the same priority on that port.

3.4.5

Frame Type Priority Override

After a frame's initial priority selection is made ([Section 3.4.4](#)) its queue priority (QPri) can be overridden either up or down based upon the frame's type. The frame's FPri cannot be modified in this way. This allows frames with a lower importance to be pushed to a lower priority inside the switch, while frames with a higher importance can be pushed to a higher priority.

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

1. Broadcast – frame's DA = FF:FF:FF:FF:FF:FF
2. PolMirror – frame is being policy mirrored – [Section 3.1.3](#)
3. PolTrap – frame is being policy trapped – [Section 3.1.3](#)
4. EType – frame's ether type matches the port's PortEType register – Port offset 0x0F
5. ARP – frame is an ARP as determined by [Section 3.3.3](#) even if ARP Mirror is disabled on the port (Port offset 0x07)
6. Snoop – frame is an IGMP/MLD as determined by [Section 3.3.4](#) and if IGMP/MLD Snoop is enabled on the port (Port offset 0x04)

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

**Note**

- The default settings in the table disable all frame type priority overrides.
- The Priority Override Table (Global 2 offset 0x0F) is also used for Secure Control on DSA ports – [Section 5.8](#).

3.4.6

Layer 2 Priority Override

After a frame's initial priority selection is made ([Section 3.4.4](#)) and after any frame type priority override is carried out ([Section 3.4.5](#)) its queue priority (QPri) and/or its frame priority (FPri) can be overridden either up or down based upon the frame's DA, SA and/or VID. This allows frames with a lower importance to be pushed to a lower priority inside and/or outside the switch, while frames of with a higher importance can be pushed to a higher priority.

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

1. VID – the frame's VLAN ID (VID) is contained in the VTU with its UseVIDPri bits set to a one ([Section 7.2.1](#)) and if the port's VTU Pri Override bits are non-zero (Port offset 0x0D)
2. SA – the frame's Source Address (SA) is contained in the ATU with any of the many Priority Override Entry States ([Section 7.3.1](#)), and if the SA is assigned to the port, and if the port's SA Pri Override bits are non-zero (Port offset 0x0D)
3. DA – the frame's Destination Address (DA) is contained in the ATU with any of the many Priority Override Entry States ([Section 7.3.1](#)), and if the DA is static, and if the port's DA Pri Override bits are non-zero (Port offset 0x0D)

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

Then if a frame's SA is contained in the ATU with a Priority Override Entry State where the entry's DPV is associated with¹ the frame's source port, the frame's QPri will be overridden to the upper two bits of the SA entry's P bits if the port's SA Pri Override register is set to a 0x2. Its FPri will be overridden to the three P bits if the port's SA Pri Override register is set to a 0x1. If the port's SA Pri Override register is set to a 0x3 then both the frame's QPri and FPri will be overridden as described above.

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

This order of processing places DA priority override above SA priority override which is above VID priority override (which is above frame type priority override – [Section 3.4.5](#)).



The default settings disable all layer 2 priority overrides.

Note

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

1.If the source port is a member of a Trunk ([Section 6.10](#)) the ATU Entry must match the source port's Trunk ID. If the source port is not a member of a trunk, the ATU Entry must have the source port's bit set to a one in its DPV.



Link Street®

88E6321/88E6320 Functional Specification

3.5

Port Based Ingress Rate Limiting (PIRL)

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

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

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

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

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

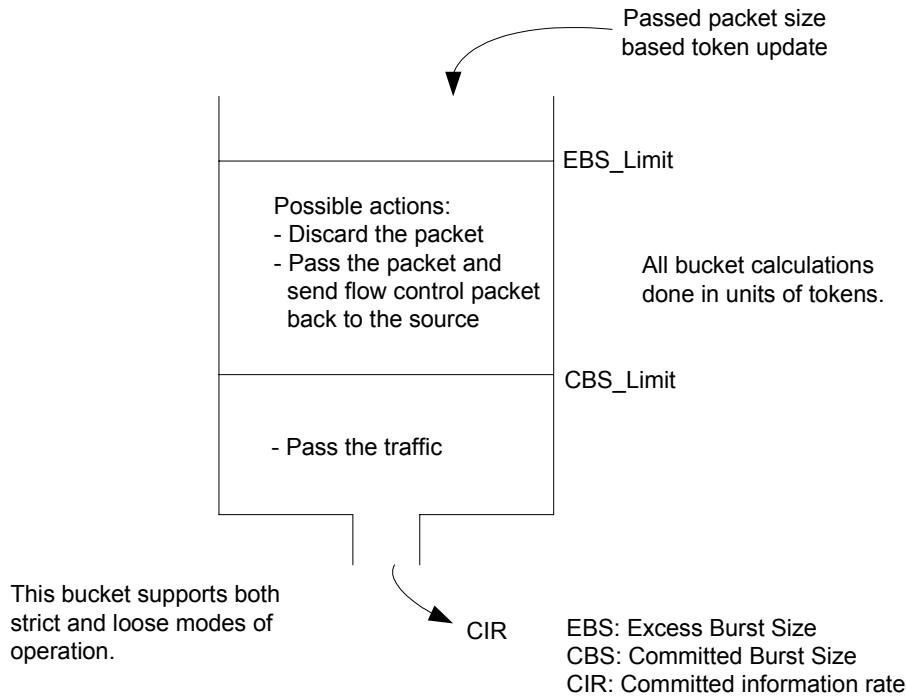
As shown in [Figure 19](#), the traffic below Committed Burst Size limit (CBS_Limit) is passed without any further actions. If the traffic burst were to continue and the bucket token depth approaches closer to the Excess Burst Size limit (EBS_Limit) by less than the CBS_Limit (i.e., EBS_Limit – CurrentBktDepth < CBS_Limit), then a programmable set of actions are specified.



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

Note

Figure 19: Color blind Leaky Bucket for Rate Limiting



As shown in Figure 19, the traffic below CBS_Limit is passed without any further actions. If the traffic burst were to continue and the bucket token depth approaches closer to the EBSLimit by less than the CBSLimit (i.e., EBSLimit – CurrentBktDepth < CBSLimit), then a programmable set of actions are specified. Note that if the frame gets discarded then the equivalent number of tokens for that frame will not get added to the bucket.

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

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

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

Another important feature that is supported by the ingress rate limiting block is Packet Sampling. In several network management applications, incoming frames from a given port may need to be sampled to a sampling port. An example would be to sample 1 out of n frames from a given port. Any

rate resource allocated for a given port can be configured to be in sampling mode. Once a rate resource is configured to be in sampling mode, it cannot be used to rate limit any packet types or priority traffic.

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

[Figure 20](#) describes a typical 88E6321/88E6320 Functional Specification device Data Rate Limiting example. [Figure 21](#) describes a typical 88E6321/88E6320 Functional Specification device Priority Based Rate Limiting example.

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

Figure 20: Data Rate Limiting Example

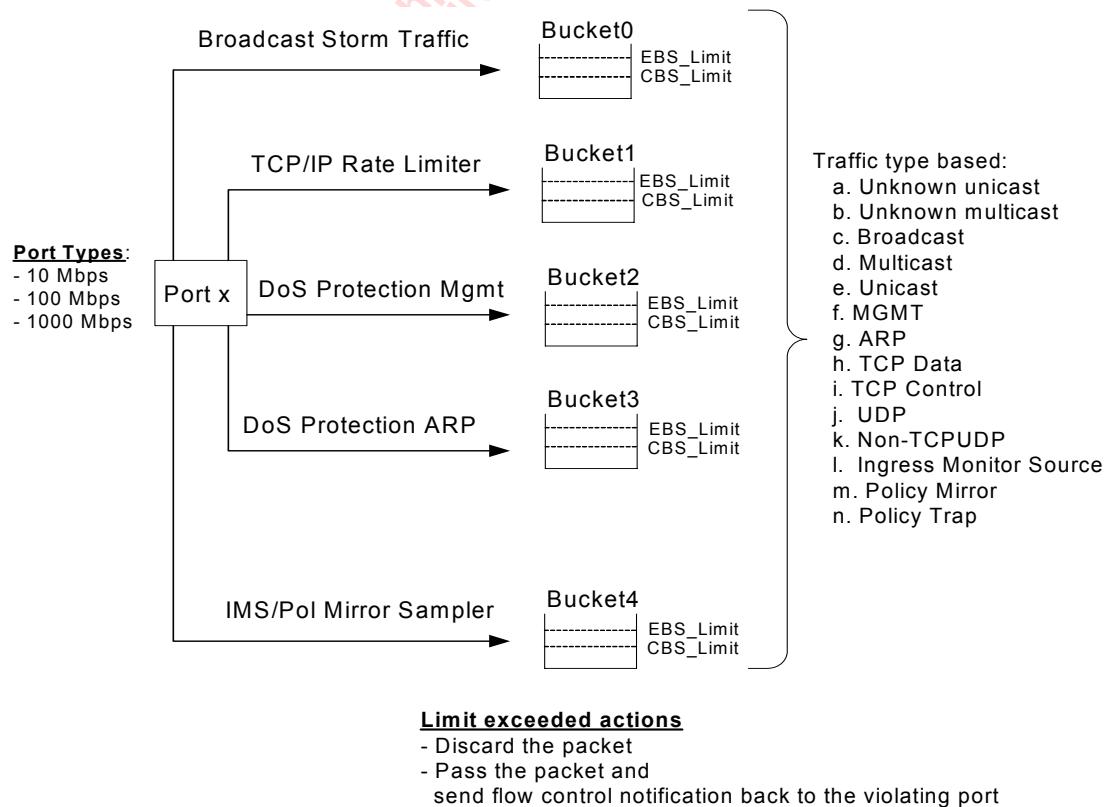
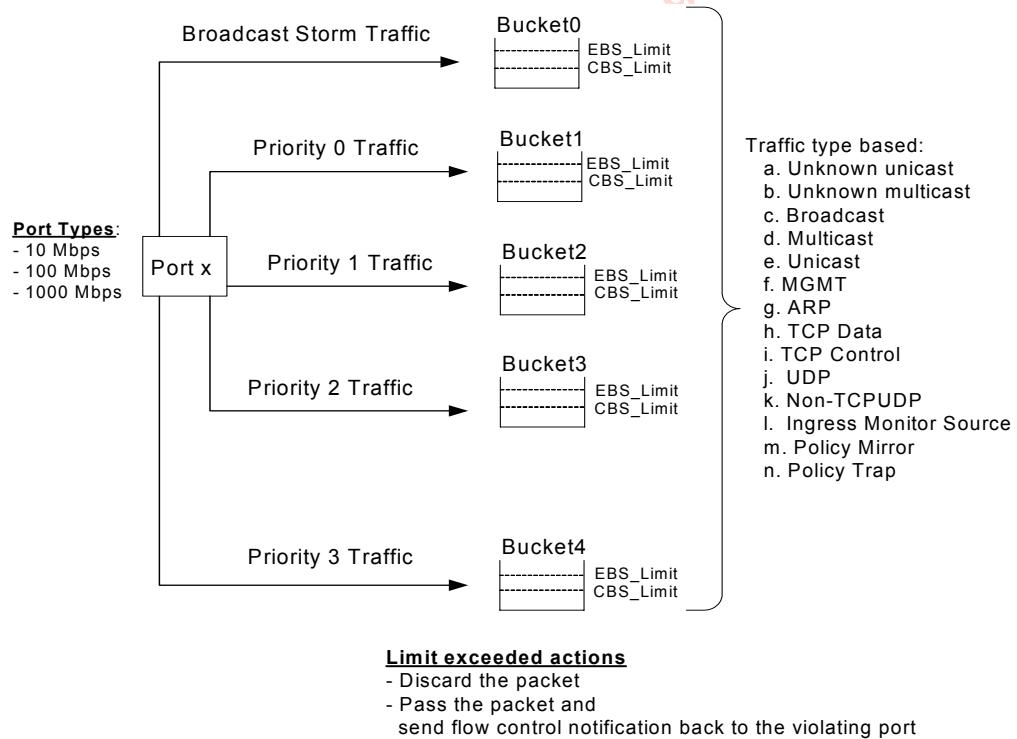


Figure 21: Priority Based Rate Limiting Example



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

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

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

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

Where the definition of:

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

Layer2 = Frame's DA to CRC

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

- TypeMask

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



Link Street®

88E6321/88E6320 Functional Specification

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

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

- Sampling Mode

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

- Account for Filtered discards

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

- Account for Queue Congestion discards

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

- PIRLFCMode

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

- PriOrPT (Priority or Packet Type)

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

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

3.6

Queue Controller

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

3.6.1

Frame Latencies

The device can guarantee frame latencies owing to its unique, high performance, four priority queuing system. A higher priority frame is always the next frame to egress a port when a port is currently egressing frames of a lower priority¹. This is true regardless of the priorities and regardless the egress port's Scheduling² mode of the switch.

3.6.2

No Head-of-Line Blocking

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

3.6.3

QoS with and without Flow Control

The Queue Controller is optimized for three modes of operation:

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

When flow control is enabled no frames are dropped and higher priority flows receive higher bandwidth through the switch (i.e., these flows are less constrained if there is congestion between a higher and a lower priority flow). The percentage of bandwidth that each flow receives is determined by the Scheduling mode see ([Section 3.6.6](#)). Flow control prevents frames from being dropped but it can greatly impact the available bandwidth on any network segment that is utilizing flow control. The latency of higher priority data on flowed-off network segments also increases since industry constrained standard flow control mechanisms stop all frames from being transmitted. Therefore, flow control may not be desirable in a QoS switch environment. For this reason, the device is also optimized to work properly without flow control.

When flow control is disabled and congestion occurs for an extended period of time, frames will be dropped. This is true in all switches. The device drops the correct frames, i.e., the lower priority frames. In this case, the higher priority flows get a higher percentage of the free buffers. The percentage of buffers they get is determined by the Scheduling mode (see [Section 3.6.6](#)).

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

1. This is true as long as the default Weighting Table is used - [Section 3.6.7](#).

2. The Scheduling mode selects a strict priority, an 8-4-2-1 weighted round robin, or a mixture of the two – [Section 3.6.7](#)

3.6.4

Guaranteed Frame Delivery without Flow Control

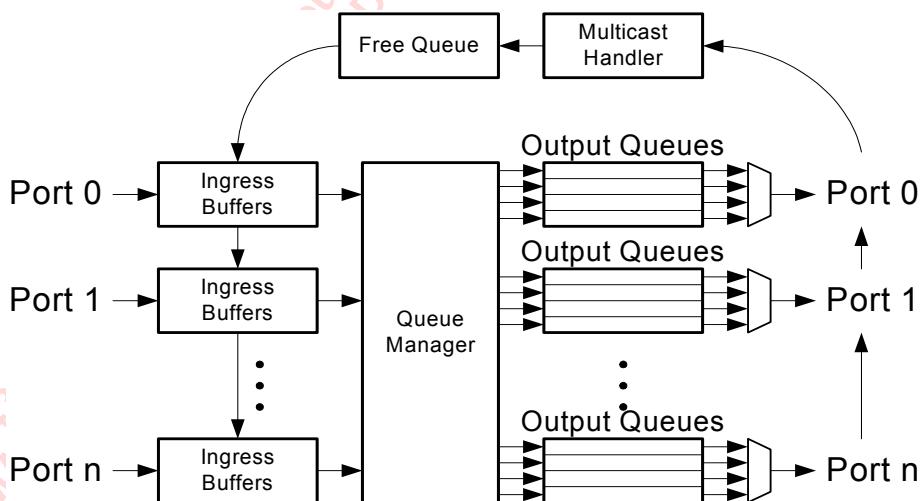
The device can guarantee high priority frame delivery¹, even with Flow Control disabled, due to its intelligent Resource Reservation system. Having an output queue with multiple priorities is not sufficient to support Quality of Service (QoS) if the higher priority frames cannot enter the switch due to a lack of buffers. The device reserves buffers for higher priority frames so they can be received and then switched. These high priority buffers are the first to get replenished from the Free Queue which prepares the receiving port for the next high priority frame.

3.6.5

The Queues

The queues in the device are shown in Figure 22.

Figure 22: Switch Queues



3.6.5.1

Queue Manager

At reset², the Queue Manager initializes the Free Queue by placing all the buffer pointers into it and ensures that all the other queues are empty. Then it takes the first available free buffer pointers from the Free Queue and assigns them to any Ingress port that is not Disabled³ and whose link⁴ is up. The switch is now ready to accept and switch packets. Whenever any port's Link goes down or if the port is set to the Disabled Port State the port's ingress Buffers and Output Queue buffers are immediately returned to the Free Queue. This prevents stale or lost buffers and allows the Free Queue to be as large as possible so larger bursts of momentary congestion can be handled. When a non-Disabled port's Link comes back up it gets its Ingress Buffers back so it can start receiving frames again.

When a MAC receives a packet it places it into the embedded memory at the address pointed to by the Input Pointers it received from the Queue Manager. When the packet is received, the MAC transfers the pointer(s) to the Queue Manager and requests new buffers from the Free Queue. If the Free Queue is empty the MAC does not receive any pointers until they become available. If the MAC starts to receive a packet when it has no pointers, the packet will be dropped. Flow control will be asserted before this occurs if it's enabled.

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

The Queue Manager uses the data returned from the Lookup Engine (see [Section 2.4.1](#)) and the Ingress Logic (see [Section 3](#) to [Section 3.4](#)) to determine which Output Queue or Queues the packet's pointer should go to and at what priority. At this point, the Queue Manager modifies the desired mapping of the frame depending upon the mode of the switch and its level of congestion. Two modes are supported, with and without Flow Control. Both modes are handled at the same time and can be different per port (i.e., one port has Flow Control enabled and another has it disabled).

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

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

3.6.5.2

Output Queues

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

Each port contains four independent Output Queues, one for each priority. The order the frames are transmitted out each port is controlled by the port's Scheduling bits (Port offset 0x0A). Strict, weighted round robin priority, or a mixture of the two can be selected (see [Section 3.6.6](#)). The weighted round robin is programmable ([Section 3.6.7](#)).

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

3.6.5.3

Multicast Handler

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

3.6.6

Per Port Fixed or Weighted Priority

The device supports strict priority, weighted round robin, or a mixture on a per egress port selection basis. The selection is made by the Scheduling bits at Port offset 0x0A. In the strict priority scheme all top priority frames egress for a port until that priority's queue is empty, then the next lower priority queue's frames egress, etc. This approach can cause the lower priorities to be starved out preventing them from transmitting any frames but also ensures that all high priority frames Egress the switch as soon as possible. In the weighted scheme an 8, 4, 2, 1 weighting is applied to the four priorities unless an alternate weighting is programmed into the QoS Weights Table ([Section 3.6.7](#)). This approach prevents the lower priority frames from being starved out with only a slight delay to the higher priority frames.



Link Street®

88E6321/88E6320 Functional Specification

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

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

3.6.7

Programmable Weighting Table

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

3.6.7.1

The Default Table

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

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

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

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

3.6.7.2

Alternate Weighting Table

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

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

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

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

While this sequence has the same weights, its ordering is very different! While this sequence allows larger bursts from the higher priorities, it does so at a latency penalty. For example, assume the current 'queue to service' pointer is pointing to the 1st '2' in the list when a priority '3' frame comes along. The priority 3 frame may have to wait behind six frames before it can egress (three more '2's, two '1's and one '0'). Keep this in mind when defining an alternate weighting.

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

3, 3, 3, 3, 2, 3, 3, 3, 3, 1, 3, 3, 3, 3, 2, 3, 3, 3, 3, 0, 3, 3, 3, 3, 2, 3, 3, 3, 3, 1, 3, 3, 3, 3, 2, 3, 3, 3, 3
(a 24, 4, 2, 1 weighting sequence with 31 steps)

3.6.7.3

Programming the QoS Weighting Table

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

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



Note

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

3.7

Embedded Memory

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

3.8

Egress Policy

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

- Non-management frame types can be blocked from leaving the switch to support Spanning Tree Protocol ([Section 3.1.1](#) for classic 802.1D and [Section 3.2.3](#) for 802.1s).
- Frames with an unknown unicast Destination Address (i.e., one not found in the address database – [Section 2.4.1](#)) can be prevented from egressing any port ([Section 3.8.1](#)).
- Frames with an unknown multicast Destination Address (i.e., one not found in the address database – [Section 2.4.1](#)) can be prevented from egressing any port ([Section 3.8.2](#)).
- 802.1Q Secure and Check Mode can filter out frames whose VID is not contained in the VLAN database ([Section 3.8.3](#)).
- Port based VLANs using the port's EgressMode bits (Port offset 0x04) and/or 802.1Q VLANs using the VID's MemberTag bits as stored in the VTU ([Section 7.2](#)) are used to determine if a frame is to be transmitted Unmodified, Tagged or UnTagged ([Section 3.8.4](#)).
- Port based Egress Rate Shaping is supported with Frames-per-Second or Layer 1, Layer 2 or Layer 3 Bits-per-Second with support for 802.3ar's Frame Overhead ([Section 3.8.5](#)).



Note Provider Tagging, otherwise known as Double Tagging or Q-in-Q Tagging is also supported. See [Section 4](#).

3.8.1

Forward Unknown/Secure Port

The device can be configured to prevent the forwarding of unicast frames with an unknown destination address (i.e., the address is not present in the address database – [Section 2.4.1](#)). The forwarding can be prevented on a per port basis by adjusting the port's EgressFloods bits in the Port Control register (Port offset 0x04) so that frames with unknown unicast addresses only go out from the port or ports where a server or router is connected.

This, together with the disabling of automatic address learning on a port ([Section 2.4.3](#)), allows a port to be configured as a Secure Port. A Secure Port allows communications to and from approved devices (by MAC address) only. In this mode all approved devices need to have their MAC address loaded as static into the address database ([Section 7.3.1](#)). When a new device tries to access the network through a Secure Port that new device's address is not automatically learned (learning must be disabled on Secure Ports) but its frame can progress to the server. When the server replies by sending a unicast frame back to the new device's address, that frame does not make it to the new device since the new device's address is not in the address database and frames with unknown unicast addresses are not allowed to egress the Secure Port. This effectively ends the communication between the un-approved new device and the rest of the network. Secure Port is similar to the 802.1X ([Section 2.4.7](#)) without the authentication server and the associated interrupts to the CPU.

3.8.2

Forward Unknown for Multicasts

The device can be configured to prevent the forwarding of multicast frames with an unknown destination address (i.e., the address is not present in the address database – [Section 2.4.1](#)). The forwarding can be prevented on a per port basis by adjusting the port's EgressFloods bits in the Port Control register (Port offset 0x04) so that frames with unknown multicast addresses only go out from the port or ports where a server or router is connected.

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

1. Load the Broadcast address into the address database so that it is known. This address would need to be loaded into each FID (Forwarding Information Database - [Section 2.4.8](#)) that requires Broadcast frame's to egress – giving control over which FIDs allow Broadcasts.
2. Or, consider frames with the Broadcast address to be special and not part of the group of other unknown multicast addresses, which is accomplished by setting the global FloodBC bit to a one (Global 2, offset 0x05). This allows frames with the Broadcast destination address to egress all ports even if the port's EgressFloods bits prevent unknown multicasts from egressing.



Note

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

3.8.3

Secure 802.1Q VLANs

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

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



Note

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

3.8.4

Tagging and Untagging Frames

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

- Secure or Check – The MemberTag bits (in VTU Data register, Global 1 offsets 0x07 to 0x09) associated with the VID assigned to the frame during ingress determines if the frame should egress unmodified, tagged or untagged (assuming egress security filtering is being performed – [Section 3.8.3](#), else Secure and Check works like Fallback below).
- Fallback – The MemberTag bits associated with the VID assigned to the frame during ingress determine if the frame should egress unmodified, tagged or untagged if the VID is contained in the VLAN database ([Section 7.2](#)). If the VID is not found in the VLAN database, or if the VID is found with the port's MemberTag bits set to 0x3, the frame egresses unmodified, tagged or untagged determined by the port's EgressMode bits (Port offset 0x04).
- 802.1Q Disabled – The port's EgressMode bits determine if the frame will egress unmodified, tagged or untagged.

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

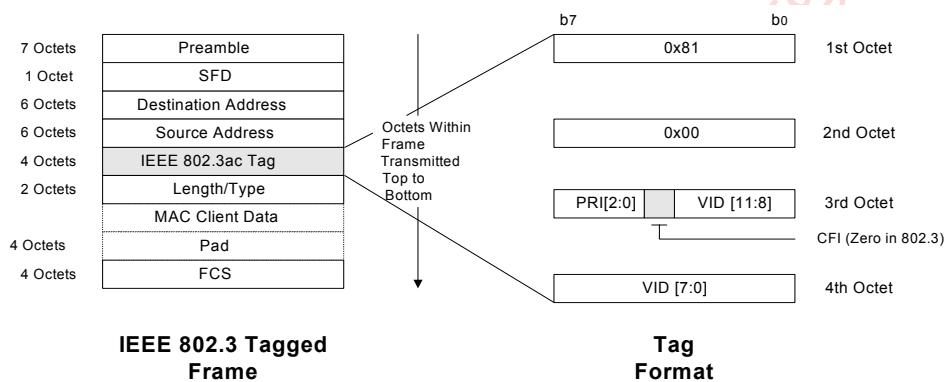
- Transmit Unmodified¹ - UnTagged frames egress the port UnTagged. Tagged frames egress the port Tagged.
- Transmit UnTagged² - UnTagged frames egress the port unmodified. The IEEE Tag on Tagged frames is removed, the frame is zero padded if needed³, and a new CRC is computed for the frame.
- Transmit Tagged⁴ - Tagged frames egress the port unmodified. An IEEE Tag is added to UnTagged frames and a new CRC is computed. The contents of the added tag is covered in [Section 3.8.4.1](#).

The format of an IEEE Tagged frame is shown in [Figure 23](#).


Note

- A physically Tagged frame is considered Tagged for the above egress frame modifications only if it enters a port where 802.1Q is enabled (i.e., the port's 802.1Q Mode in Port offset 0x08 is Secure, Check or Fallback). If a switch is configured where some ports have 802.1Q enabled (using Q VLANs) and some ports have 802.1Q disabled (using Port Based VLANs) extreme care is required to prevent physically Tagged frames from egressing double tagged (because physically Tagged frames are considered UnTagged for egress frame modifications if they enter a port where 802.1Q is Disabled). The best way to prevent this is set all port's EgressMode bits (Port offset 0x04) to Transmit Unmodified. This ensures frames are transmitted looking the way they did when they entered the switch unless the egress port's 802.1Q is Enabled and the frame's VID is in the VTU. Make sure the DefaultVID (Port offset 0x07) on ports where 802.1Q is Disabled is not contained in the VTU, or if it is, make sure its MemberTag bits are set to Transmit Unmodified.
- If a Tagged frame is Transmitted Tagged, its VID is updated to be the VID assigned during Ingress ([Section 3.2.2.5](#)) and its PRI bits are updated to be the FPri assigned during Ingress ([Section 3.4](#)).

Figure 23: IEEE Tag Format



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

3.8.4.1

Adding a Tag to Untagged Frames

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

- The 1st Octet is always 0x81
- The 2nd Octet is always 0x00
- The PRI bits indicate the frame's priority (FPri) determined during Ingress ([Section 3.4](#))
- The CFI bit is always set to a zero
- The VID bits indicate the VID assigned to the frame during Ingress ([Section 3.2.2.5](#))



A Tag that is added due to Provider Tagging is done differently. See [Section 4](#).

Note

3.8.4.2

Priority Re-Map and Priority Override

When a Tagged frame egresses a port Tagged, the PRI bits in the tag are modified to reflect the frame's priority (FPri) that was determined during Ingress ([Section 3.4](#)). The PRI bits can be re-mapped by the ingress port's IEEE Priority Remapping registers (Port offsets 0x18 and 0x19) or the frame's PRI bits can be ignored and the ingress port's default priority used instead, or the frame's priority can be overridden.



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

Note

3.8.4.3

VID 0x000 and VID Override

A Tagged frame egressing a port Tagged may have its VID bits modified. If the Ingressing frame's VID was 0x000, or if the Ingress port's DefaultVID (Port offset 0x07) is assigned to the frame instead. See [Section 3.2.2.5](#).



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

Note



Link Street®

88E6321/88E6320 Functional Specification

3.8.5

Egress Rate Shaping

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

The supported modes are:

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

3.8.5.1

Frames-per-Second Egress Shaping

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

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



Note

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

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

- $\text{EgressRate} = \text{EgressDec} / (32 \text{ ns} * \text{Desired Egress Frame Rate per Second})$



Note

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

- For Example: Set the egress rate to 7700 frames-per-second
- $\text{EgressRate} = \text{EgressDec} / (32 \text{ ns} * 7700 \text{ frames-per-second})$
- $\text{EgressRate} = 1 / (0.00000032 * 7700)$
- $\text{EgressRate} = 4058.44$ or 0xFDA
- $\text{EgressDec} = 0x01$

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

- $\text{ActualRate} = \text{EgressDec} / (32 \text{ ns} * \text{EgressRate})$
- $\text{ActualRate} = 1 / (0.00000032 * 4058)$
- $\text{ActualRate} = 7700.84$ frames per second



Note

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

3.8.5.2

Bits-per-Second Egress Shaping

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

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



Note

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

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

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

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

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

$$\text{EgressRate} = 8 \text{ bits} * \text{EgressDec} / (32\text{ns} * \text{Desired Egress Bit Rate per Second})$$



Note

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

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

- EgressRate = 3906.25 or 0xF42 (see NOTES above)
- EgressDec = 0x04 (256 k / 64 k = 4)
- Due to the cropping of fractions, the actual rate will be:
- ActualRate = 8 bits * EgressDec / (32 ns * EgressRate)
- ActualRate = 32 / (0.000000032 * 3906)
- ActualRate = 256.016 bits-per-second

**Note**

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

3.8.5.3

Frame Overhead Egress Shaping

When a port is configured in one of the Bits-per-Second egress rate shaping modes ([Section 3.8.5.2](#)) an optional overhead count, which follows the IEEE 802.3ar proposals, can be added to each frame. This fixed overhead is designed to adjust the egress rate taking into account frame modifications that may occur upstream from this device, such as adding a tag to the frame.

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

4

Provider Mode Ports

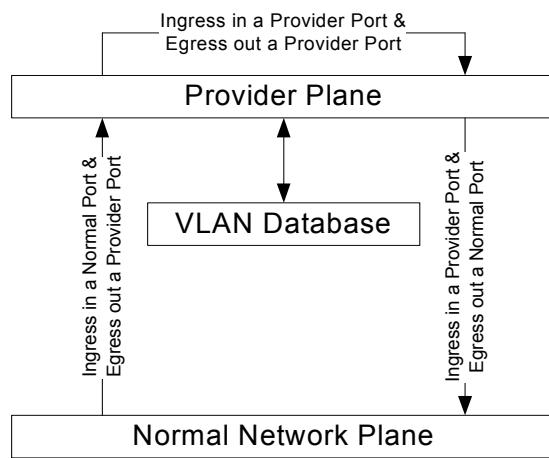
The discussions that follow assume that Provider ports are set in Provider mode (i.e., FrameMode = 0x2 at Port offset 0x04) and that Customer ports are set in Normal Network mode (i.e., FrameMode = 0x0 at Port offset 0x04). Provider mode is sometimes referred to as Double Tagging, Q-in-Q, or IEEE 802.1ad.

Provider ports are designed to bridge between Customer ports and Provider ports and visa versa. More than one Provider port can be supported (in a single device or between multiple devices interconnected with DSA ports – [Section 5](#)).

**Note**

- When one or more Provider ports exist on the switch, Customer ports are expected to be Normal Network ports as defined in [Section 3](#) although 802.1Q should not be enabled on Customer ports. Instead Customer ports should use the Port Based VLAN mode if port to port isolation is required for Customer to Customer data flow separation ([Section 3.2.1](#)). The Customer ports' EgressMode should also be set to Transmit Unmodified (Port offset 0x04 and [Section 3.8.4](#)). These restrictions occur because the 802.1Q VLAN Database is being used by the Provider Port(s) for S-TAG (Service Tag) switching and therefore it is not available for C-TAG (Customer Tag) switching and dynamic egress tag modifications. Per VLAN Spanning Tree (IEEE 802.1s – [Section 3.2.3](#)) is not available on the Customer ports for the same reason (the VTU and STU are being used by the Provider port(s)).
- Provider Ports must have their EgressMode set to 0x0 (transmit unmodified, Port offset 0x04).

Figure 24: Provider vs. Normal Data Planes

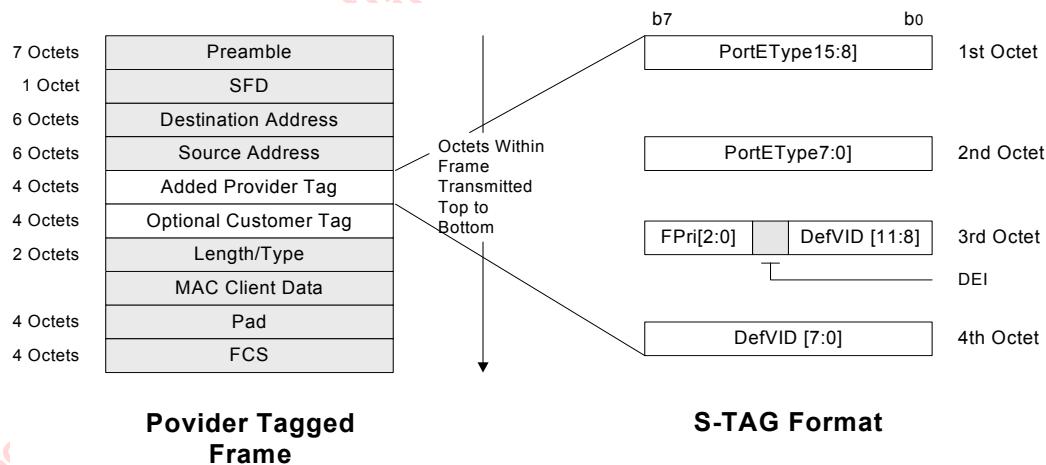


4.1 Customer to Provider

As stated in the NOTE above, Customer ports are expected to have their 802.1Q Mode Disabled (Port offset 0x08). This causes the ingress port's DefaultVID (Port offset 0x07) to be assigned to all ingressing frames as the Customer's S-VID regardless if the Customer's frame is tagged or untagged. The frame is then port based VLAN limited (by software configuration – [Section 3.2.1](#)) to egress only the correct ports, for example, only to the Provider Port(s). All the other Normal Network features of the device (as described in [Section 2](#) and [Section 3](#)) are available to Customer ports, including Layer 2 PCLs ([Section 3.1.3](#)), QoS mapping ([Section 3.4](#)) and ingress rate limiting ([Section 3.5](#)), to name a few.

When the Customer's frame egresses a Provider Port the frame will egress with an S-TAG (service tag) added in front of the Customer's Tag, if one existed in the frame ([Figure 25](#)).

Figure 25: Provider Tag Format



The Ether type of the added S-TAG (1st and 2nd octet) comes from the egress port's PortEType register (Port offset 0x0F). The PRI bits (3rd octet) comes from the FPri assigned to the frame during ingress ([Section 3.4](#)) and the DEI (Drop Eligible Indicator) is set to zero. The S-VID (3rd and 4th octets) is set to the VID assigned to the frame during ingress, which will be the source port's DefaultVID (Port offset 0x07) if 802.1Q Mode is Disabled on the source port¹ (Port offset 0x08).



Note

The action described above occurs the same way regardless if the Customer port and Provider port are in the same physical switch device, or if they are in separate switch devices interconnected with DSA ports ([Section 5](#)).

1. Or it will be the S-VID from the frame's S-TAG if the came in another Provider Port (see [Section 4.4](#)).

4.2

Provider to Customer

When a frame enters a Provider Port, its Ether type (1st and 2nd octet in [Figure 25](#)) is compared to the Provider port's PortEType register (Port offset 0x0F). If a match exists, the frame is considered Provider Tagged (i.e., it has an S-TAG) and the frame's S-VID (3rd and 4th octets) is assigned as the frame's VID inside the switch (this cannot be overridden by the Provider port's ForceDefaultVID bit, [Section 3.2.2.6](#), as a Provider Port is trusted – but read on below how the ForceDefaultVID bit may be used on Provider Ports). The ether type match also causes the frame's S-PRI (3rd octet) bits to be assigned as the frame's initial FPri. The 4-byte S-TAG is then removed from the frame (during ingress), getting it ready to be transmitted out a Customer port¹. A new CRC is placed on the frame due to the removal of the S-TAG². The frame is stored in the switch's memory looking identical to the way the Customer sent it into the switch, i.e., it is unmodified as far as the customer is concerned.

If the ingressing frame's ether type does not match the Provider port's PortEType register, the frame is not considered Provider Tagged. In this case the frame is processed as if it entered a Normal Network port getting a VID, FPri and QPri assigned to it as defined in [Section 3](#), with one exception. The Provider port's DiscardTagged and DiscardUntagged bits (Port offset 0x08) work differently as follows:

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

The DiscardUntagged function on Provider Ports can be used to ensure only 'good' Provider Tagged frame are allowed to enter the Provider Port. If however, non-Provider Tagged frames are allowed to enter the switch (by clearing the port's DiscardUntagged bit to zero) the Customer port(s) the frame is allowed to egress can be limited by setting the Provider Port's ForceDefaultVID bit to a one (Port offset 0x07) and setting the port's DefaultVID to an unused value. A DefaultVID of 0x000 will work here as the VID of 0x000 exists in the VTU just like all the other VID values. The ForceDefaultVID feature ([Section 3.2.2.6](#)) will only take effect on a Provider Port if the ingressing frame is not considered Provider Tagged (i.e., its ether type does not match the port's PortEType register). This DefaultVID assigned to the non-Provider frames will not show up in the frame as long as the MemberTag bits for this VID are set to egress unmodified (see [Section 4.2.1](#)).

4.2.1

Provider VID Processing

The VID assigned to the frame (the S-TAG's S-VID or the port's DefaultVID in case the frame was not properly Provider Tagged, see above) is used to access the VLAN Database assuming 802.1Q Mode (Port offset 0x08) is enabled on the Provider Port ([Section 3.2.2](#)). VLAN switching is needed to get each provider frame mapped to the correct Customer Port(s) while at the same time ensuring the frame does not get to any of the wrong Customers. The VTU's MemberTag bits for the Customer port(s) for each VID entry should be set to either:

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

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

4.2.2

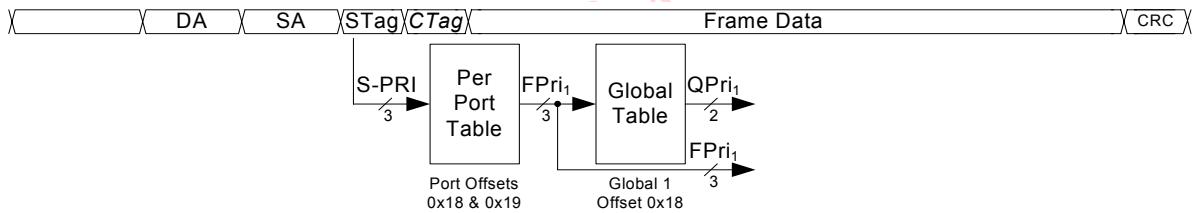
Provider QoS Processing

When the ingressing frame is considered Provider Tagged (i.e., its ether type matches the port's PortEType register) its S-PRI bits from the frame's removed STAG are used to define FPri1 and QPri1 as shown in [Figure 26](#) below. This no different from the way Normal Network ports work ([Section 3.4.1](#)) i.e., the frame's 1st Tag is used. This means that any data from the frame's C-TAG

1. Frames are padded back up with zeros just before the CRC if the frame is less than 64 bytes in size due to the removal of an S-TAG.
2. Ingressing frames with a CRC error are discarded, therefore a bad frame cannot be made good by adding the new CRC.

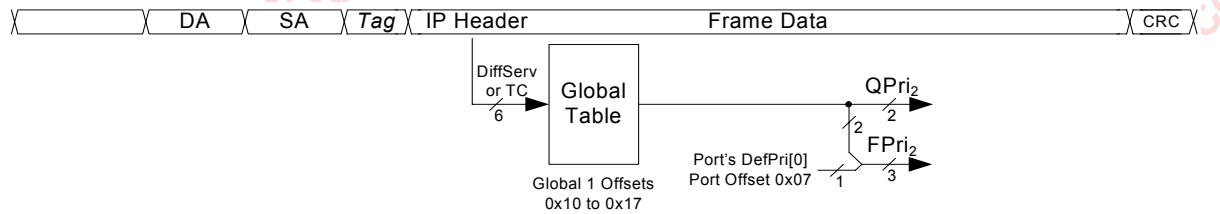
(the Customer's Tag that follows the S-TAG, if one exists) cannot be used for QoS determination. It is assumed the S-Tag's priority can be trusted as it came from the Provider, and this is the proper priority to use inside this switch as this switch is still 'owned' by the Provider, giving service to many Customers. The only way to guarantee this service is to control it.

Figure 26: Provider S-PRI Remapping



Since the S-TAG is removed during ingress, the IP portion of the frame can still be used to determine FPri₂ and QPri₂ as shown in [Figure 27](#). This is not recommended unless the IP priorities can be trusted.

Figure 27: Provider IP PRI Mapping



The rest of the QoS selection on Provider ports is the same as described in [Section 3.4](#) (i.e., Initial Priority Selection, Frame Type Priority Overrides and Layer 2 Priority Overrides can be done).

If the ingressing frame is not considered Provider Tagged (i.e., the frame's Ether type does not match the Provider port's PortEType register) the QoS of the frame is determined as if it entered a Normal Network port getting FPri and QPri assigned to it as defined in [Section 3.4](#).

4.3

Customer to Customer

Multiple Customer ports owned by the same Customer can be supported. If any single customer has more than one port¹ on the Provider switch, frame traffic between these Customer ports can occur directly, without the need of sending these frames to the Provider. This can be accomplished by expanding the Customer ports' Port Based VLANs (both in-chip and cross-chip – [Section 3.2.1](#) and [Section 5.5.3](#)) to include all ports owned by the same Customer. By assigning all ports going to the same Customer with the same DefaultVID (Port offset 0x07) and the same FID ([Section 2.4.8](#)), the address database can be used to direct the frames to the correct port(s).

Any QoS priority overriding (including FPri changes) that was done inside the switch for frames going to the Provider port will not take effect outside the switch when the frame goes to another Customer port, assuming all Customer port's are configured to transmit frames unmodified (as defined by the Customer port's EgressMode bits, Port offset 0x04 for Customer to Customer flows, and by the MemberTag bits for the Customer's port as loaded into the VTU for the Customer's S-VID, [Section 4.2](#), for Provider to Customer flows). This is the proper operation of a provider switch, i.e., to provide a 'pipe' for Customer data to flow through without the Customer data getting modified in any way.

Dynamic 802.1Q VLAN switching, i.e., limiting which ports the frame can go out, based upon the frame's Tag (C-Tag in this case) cannot be done for the Customer to Customer flows as the VLAN database is needed by the Provider Port(s) and its not available for the Customer ports (see [Figure 24](#)). As stated in [Section 4.1](#), 802.1Q Mode should be disabled on Customer ports.

4.4

Provider to Provider

More than one Provider port going to the same or different Providers is supported. If any single Provider has more than one port² on the Provider switch, frame traffic between these Provider ports can occur directly, with dynamic 802.1Q VLAN switching if needed. This is accomplished automatically (both in-chip and cross-chip). During ingress, the Provider's S-TAG is removed from the frame (as defined in [Section 4.2](#)), with its S-VID being assigned as the frame's VID. If this frame is mapped to another Provider port, that port will insert a new Provider S-TAG as the frame egresses following the rules defined in [Section 4.1](#). In this case the VID assigned during ingress will be the S-VID extracted from the frame when it ingressed the Provider Port.

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

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

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

The VID assigned to the frame (the S-TAG's S-VID or the port's DefaultVID in case the frame was not properly Provider Tagged, see [Section 4.2](#)) is used to access the VLAN Database assuming 802.1Q Mode (Port offset 0x08) is enabled on the Provider Port(s) ([Section 3.2.2](#)). VLAN switching is needed to get each provider frame mapped to the correct Provider and/or Customer Port(s) while at the same time ensuring the frame does not get to any of the wrong Customers and/or Providers.

1. This discussion is for non-Link Aggregated ports. Link Aggregated ports to a Customer are also supported using the standard rules for Port Trunks ([Section 6.10](#)).
2. This discussion is for non-Link Aggregated ports. Link Aggregated ports to a Provider are also supported using the standard rules for Port Trunks ([Section 6.10](#)).



Link Street®

88E6321/88E6320 Functional Specification

The VTU's MemberTag bits for any alternate Provider port(s) for each VID entry should be set to either:

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

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

4.5

Recursive Provider Tag Stripping

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

Recursive Provider tag stripping removes one or more valid S-TAGs found in a frame as it ingresses a Provider port. A valid S-TAG is any S-TAG that has an Ether type that matches the port's PortEType register. The VID and FPri assigned to the frame for switch processing ([Section 4.2](#)) always comes from the first, or outer, S-TAG that is removed. The content of all subsequent S-TAGs is ignored. Frames are padded back up with zeros just before the CRC if the frame is less than 64 bytes in size due to the removal of an S-TAG¹.

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

4.6

Restrictions on Provider Ports

All the Normal Network port functions described in [Section 2](#) and [Section 3](#) are available to Provider ports with the exception of any function that uses the port's PortEType register (as this register is needed to determine the Provider frame's Ether type on Provider Ports). Specific functions that should not be used on Provider ports:

- Don't use Layer 2 Policy Controls using the frame's Ether type ([Section 3.1.3.2](#)).
- Don't use Frame Type Priority Override using the frame's Ether type ([Section 3.4.5](#)).
- Don't use a Transmit Tagged or Transmit Untagging Egress Mode. Always use Transmit Unmodified² ([Section 3.8.4](#)).
- Don't used a Provider Port as a destination port for Mirrors³ ([Section 3.1.3](#) for Layer 2 Mirrors and [Section 6.9](#) for Ingress and/or Egress Mirrors).

4.7

Restrictions on Customer Ports

All the Normal Network port functions described in [Section 2](#) and [Section 3](#) are available to Customer ports with the exception of any function that uses the VTU (as the VTU's contents are needed for the Provider ports, see [Section 4.2](#)). Specific functions that should not be used on Customer ports when one or more Provider ports exist on the switch are:

- Don't enable any of the 802.1Q VLAN modes ([Section 3.2.2](#)). Use Port Based VLANs only.
- Don't use 802.1s Per VLAN Spanning Tree ([Section 3.2.3](#)).
- Don't use Layer 2 Priority Override using the frame's Customer VID ([Section 3.4.5](#)).
- Don't use a Transmit Tagged or Transmit Untagging Egress Mode. Always use Transmit Unmodified⁴ ([Section 3.8.4](#)).

1. Ingressing frames with a CRC error are discarded, therefore a bad frame cannot be made good by adding the new CRC.
2. Provider ports get an S-TAG added automatically based upon FrameMode (Port offset 0x04) instead of using any other indicator. Therefore, Transmit Unmodified will do what is needed and is the correct EgressMode to use.
3. Mirrors cause a frame to be replicated, the original frame and then a mirrored or copied control frame. If these copied frames egress a Provider port there is no way for the provider to distinguish which frame was the original frame and which one was the copy. But a management CPU can. An alternate approach is to use a dedicated port for the mirrored data that contains only the copies (see [Section 6.9](#)).
4. See [Section 4.2.1](#) for the reasons why Transmit Unmodified is the correct EgressMode to use on Customer ports.

5

Distributed Switch Architecture (DSA) Ports

The discussions that follow assume that ports used to interconnect devices to each other and to the management CPU (referred to as internal ports) are set in Distributed Switch Architecture (DSA) Tag mode (i.e., FrameMode = 0x1 at Port offset 0x04, except for Ether type DSA Tag, [Section 5.9](#), where FrameMode = 0x3) while external ports are set to either Normal Network mode ([Section 3](#)) or Provider mode ([Section 4](#)).

An alternate DSA Tag mode, called Ether type DSA Tag ([Section 5.9](#)) is also supported. Ether type DSA encapsulates the standard DSA Tag after a programmable Ether type. The Ether type DSA mode is optimized for switch to CPU interconnections while the standard DSA is optimized as a chip-to-chip switch fabric extension.

Standard DSA Tag ports must have their EgressMode set to 0x0 (transmit unmodified, Port offset 0x04). Ether type DSA Tag ports can use any EgressMode setting ([Section 5.9](#)).

There are four major DSA Tag mode frame types:

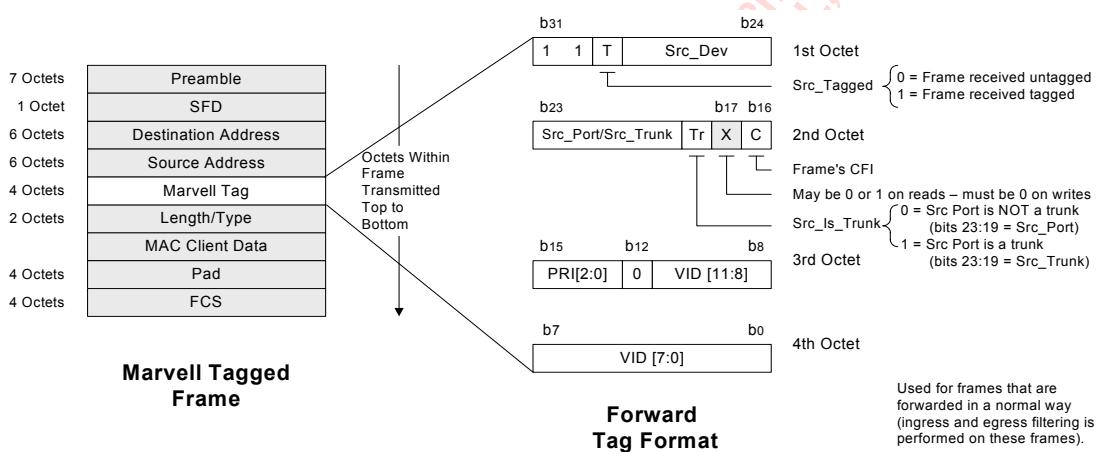
- Forward – for normal frames
- To_CPU – for MGMT (management) or control frames that needs to go to the CPU
- From_CPU – for MGMT (management) or control frames that come from the CPU
- To_Sniffer – for MGMT (management) or control frames used for chip-to-chip communication

5.1

Forward DSA Tag

The Forward DSA Tag is applied to a frame as it egresses a DSA Tag port if the frame is not a special frame (e.g., it is not a MGMT frame – [Section 6.3](#)). It is the kind of DSA Tag that a CPU should use when sending a frame into the switch where the CPU wants the switch to process the frame and figure out where it should go. Normal ingress and egress filtering rules apply to these frames, i.e., the frames are processed as if they entered a Normal Network port ([Section 2](#) and [Section 3](#)). Most of the frames that go through a DSA Tag port contain the Forward DSA Tag format, defined in [Figure 28](#) and [Table 11](#).

Figure 28: Forward DSA Tag Format





Link Street®

88E6321/88E6320 Functional Specification

Table 11: Forward DSA Tag Fields

Frame's Field	Description
Src_Tagged	Source Tag Mode, i.e., the DSA Tag is placed in the frames on top of the standard IEEE Tag that was in the frame. Standard IEEE Tags contain an 0x8100 Ether type. This bit can only be set to a one if the frame came from a Normal Network port (Section 3) where 802.1Q is enabled (Section 3.2.2) and if the frame was IEEE Tagged. This bit will never be set to a one if the frame came in on a Provider Port (Section 4) nor if it entered a Normal Network port where 802.1Q Mode is Disabled even if the frame is IEEE Tagged (i.e., Port Based VLANs are being used). In these cases the DSA Tag is added to the frame leaving the rest of the frame's contents intact. This bit is used to tell the egress logic on Normal Network ports how to convert the frame from DSA Tag to Normal Network (see Table 12). The bit is ignored if the frame egresses a Provider Port as a DSA Tag is always converted into a Provider Tag with the Provider port's PortEType register being used as the Tag's Ether type ¹ .
Src_Dev	Source Device. These bits are used to define the original source device's number where the frame first ingressed (i.e., the first device where the frame Ingressed from a Normal Network (Section 3) or Provider port (Section 2) before being switched to an Internal DSA Tag port). These bits come from the source device's DeviceNumber register (Global 1 offset 0x1C).
Src_Port/Src_Trunk	Source Port or Source Trunk. If the Src_Is_Trunk bit, below, is zero, these bits are used to define the original source port's number (on the source device above). 0x00 indicates Port 0, 0x01 indicates Port 1, 0x02 for Port 2, 0x03 for Port 3 etc. If the Src_Is_Trunk bit, below, is one, these bits are used to define the Trunk ID of the 1st trunk this frame entered or passed through.
Src_Is_Trunk	Source is Trunk. When this bit is zero, it indicates this frame originally entered a non-trunked port and this frame has never passed through a trunked port. In this case, the Src_Port/Src_Trunk bits define the Src_Port. When this bit is one it indicates this frame originally entered a trunked port or this frame passed through a DSA port that was trunked. In this case, the Src_Port/Src_Trunk bits define the Src_Trunk.
C	The original frame's CFI (Canonical Format Indicator) bit if the frame was IEEE tagged when it originally entered a Normal Network port (Section 3) on the switch. It is the original frame's DEI (Drop Eligible Indicator) bit if the frame was Provider tagged when it originally entered a Provider port (Section 2) on the switch.
PRI[2:0]	The frame's FPP priority as determined by the Ingress rules of the last devices that this frame entered (Section 3.4 for Normal Network ports and Section 4.2.2 for Provider ports).
VID[11:0]	The frame's VLAN identifier as determined by the Ingress rules of the last devices that this frame entered (Section 3.2.2.5 for Normal Network ports and Section 4.2.1 for Provider ports).

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

Table 12: Src_Tagged vs. Normal Network Egress Mode Actions

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

1. As defined in [Section 3.8.4](#).



Note

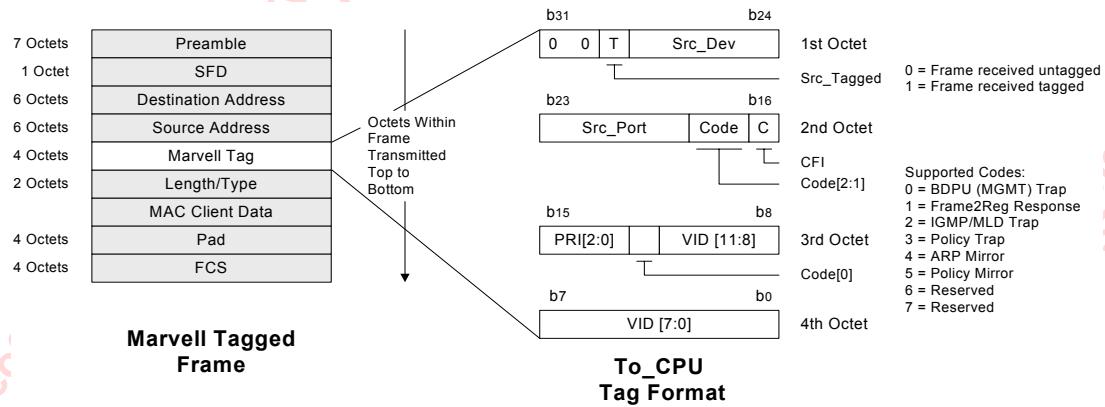
When the CPU sends a frame into the local switch device using the Forward DSA Tag format, it should set the Src_Dev to the value of the DeviceNumber (Global 1 offset 0x1C) of the device it is attached to and it should set the Src_Port equal to the port number on the device it is attached to. Bit 17 and bit 12 in [Figure 28](#) must be cleared to zero. This way the Src_Dev and Src_Port values will match the physical port where the frames entered the switch. This will make all the features that used these fields in the frame more consistent (like cross-chip Port Based VLANs – [Figure 5.5.3](#)).

5.2 To_CPU DSA Tag

When the CPU is running the Spanning Tree Protocol ([Section 6.2](#)) or it needs the switch to perform IGMP/MLD Snooping ([Section 3.3.4](#)), ARP Mirroring ([Section 3.3.3](#)) or Layer 2 Policy ([Section 3.1.3](#)), the CPU must receive some special frames. When the switch device is configured to detect special frames and they are also configured to egress these frames from a DSA Tag port (like to the CPU's port), the frame is modified with a To_CPU DSA Tag as it egresses the port. The format of the To_CPU Tag is defined in [Figure 29](#) and [Table 13](#).

In an environment where more than one device is connected together using DSA Tag ports, the device(s) in the middle will receive To_CPU DSA Tag frames. These frames are sent through the intermediate switch unmodified. They egress from the port defined by the CPUDest register (Global 1 offset 0x1A) unless the frame's CPU Code = 0x5 where it will egress from the port defined by the MirrorDest register instead (also in Global 1 offset 0x1A). This allows Layer 2 Policy Mirrors ([Section 3.1.3](#)) to be directed to some other port from the one the management CPU is connected to. In this way, a Normal Network port can be the final destination for Layer 2 Policy Mirrors.

Figure 29: To_CPU DSA Tag Format



Note

- If a CPU sends a To_CPU DSA Tag frame into the switch, it will be mapped as stated above, which means the CPU will likely receive the frame back unmodified (depending upon the frame's CPU Code and the setting of the CPUDest and MirrorDest registers). This can be used as a diagnostic to test the flow of frames between the CPU and the switch. If the CPU needs to send frames out a specific port, use the From_CPU DSA Tag frame format instead.
- To_CPU DSA Tag frames are considered MGMT (management) frames. MGMT frames are processed differently inside the switch. See [Section 5.6](#).

Table 13: To_CPU DSA Tag Fields

Frame's Field	Description
Src_Tagged	Source Tag Mode, i.e., the DSA Tag is placed in the frames on top of the standard IEEE Tag that was in the frame. Standard IEEE Tags contain an 0x8100 Ether type. This bit can only be set to a one if the frame came from a Normal Network port (Section 3) where 802.1Q is enabled (Section 3.2.2) and if the frame was IEEE Tagged. This bit will never be set to a one if the frame came in on a Provider Port (Section 4) nor if it entered a Normal Network port where 802.1Q Mode is Disabled even if the frame is IEEE Tagged (i.e., Port Based VLANs are being used). In these cases the DSA Tag is added to the frame leaving the rest of the frame's contents intact.
Src_Dev	Source Device. These bits are used to define the original source device's number where the frame first ingressed (i.e., the first device where the frame Ingressed from a Normal Network (Section 3) or Provider port (Section 4) before being switched to an Internal DSA Tag port). These bits come from the source device's DeviceNumber register (Global 1 offset 0x1C).
Src_Port	Source Port. These bits are used to define the original source port's number (on the source device above). 0x00 indicates Port 0, 0x01 indicates Port 1, 0x02 for Port 2, 0x03 for Port 3 etc. These bits always reflect the physical Src_Port of To_CPU frames even if the physical Src_Port is a Trunk port (Section 6.10).
Code	To_CPU frame type code. These bits are set by the original Src_Dev (see above) to indicate the kind of To_CPU frame. The device generates a frame type code depending upon the reason as defined in Table 14 .
C	The original frame's CFI (Canonical Format Indicator) bit if the frame was IEEE tagged when it originally entered a Normal Network port (Section 3) on the switch. It is the original frame's DEI (Drop Eligible Indicator) bit if the frame was Provider tagged when it originally entered a Provider port (Section 4) on the switch.
PRI[2:0]	The frame's FPri priority as determined by the Ingress rules of the last devices that this frame entered (Section 3.4 for Normal Network ports and Section 4.2.2 for Provider ports).
VID[11:0]	The frame's VLAN identifier as determined by the Ingress rules of the last devices that this frame entered (Section 3.2.2.5 for Normal Network ports and Section 4.2.1 for Provider ports).

Table 14: To_CPU Code Meanings

Code	Name	Comments
0x0	MGMT Trap	Placed on re-directed (trapped) frames that come from DA MGMT Trapping (Section 6.3)
0x1	Frame2Reg	Placed on Remote Management response frames (Section 8)
0x2	IGMP/MLD Trap	Placed on re-directed (trapped) frames that come from IGMP/MLD Trapping (Section 3.3.4)
0x3	Policy Trap	Placed on re-directed (trapped) frames that come from Layer 2 Policy Traps
0x3	Reserved	Reserved for future use.
0x4	ARP Mirror	Placed on mirrored or copied frames that come from ARP Mirroring (Section 3.3.3)
0x5	Policy Mirror	Placed on mirrored or copied frames that come from Layer 2 Policy Mirrors
0x5	Reserved	Reserved for future use.
0x6	Reserved	Reserved for future use.
0x7	Reserved	Reserved for future use.

5.3

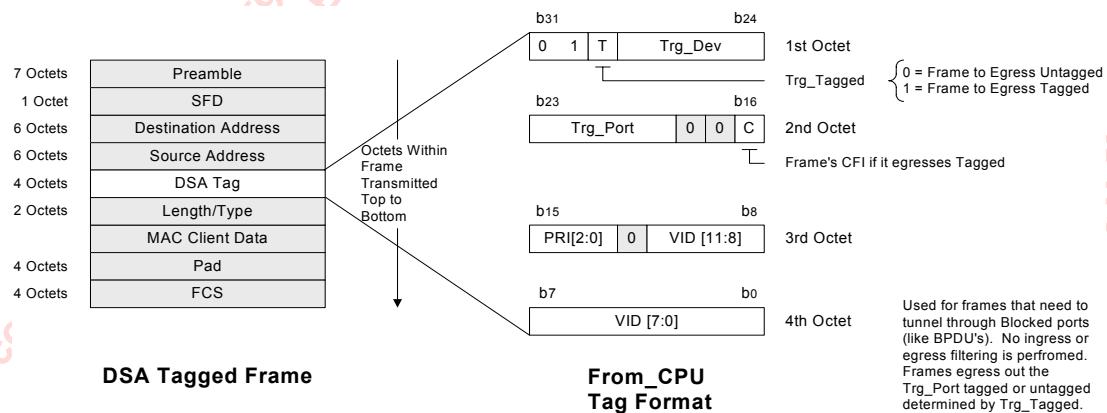
From_CPU DSA Tags

When the CPU is running the Spanning Tree Protocol ([Section 6.2](#)), or other protocols, it needs to be able to transmit special frames, like BPDU's, out any of the ports on the switch. The CPU can do this by using the From_CPU DSA Tag. The format of the From_CPU Tag is defined in [Figure 30](#) and [Table 15](#).

In an environment where more than one device is connected together using DSA Tag ports, the device(s) in the middle will receive From_CPU DSA Tag frames. In this case, the receiving port examines the frame's Target Device (Trg_Dev) field to see if this device is the target (by comparing the frame's Trg_Dev value to the local device's DeviceNumber register at Global 1 offset 0x1C). If this is the target device, the frame is sent out the port indicated in the frame's Target Port (Trg_Port) field. In this case, it is expected that the frame will be mapped to a Normal Network ([Section 3](#)) or a Provider port ([Section 4](#)) where the frame will egress the port IEEE Tagged or Untagged based on the frame's Trg_Tagged bit (see [Table 15](#)).

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

Figure 30: From_CPU DSA Tag Format



Note

- From_CPU DSA Tag frames are considered MGMT (management) frames. MGMT frames are processed differently inside the switch – including the ability to egress Blocked ports ([Table 5](#)). See [Section 5.6](#).
- The CFI bit in From_CPU DSA Tag frames is placed at b16. When this frame egresses out a normal network port the CFI bit will be moved to its correct position at b12 (which is the DEI bit on Provider Ports – [Section 4](#)).

Table 15: From_CPU DSA Tag Fields

Frame's Field	Description
Trg_Tagged	Target Tag Mode. This bit allows the CPU to define if this frame is to egress the target port IEEE Tagged or not. If this bit is set to a one, the frame egresses the target port with a proper IEEE 802.3ac Tag (i.e., the 1st two octets of the From_CPU DSA Tag are converted to the 0x8100 Ether type and the frame's C bit is moved between the PRI and VID bits). If this bit is cleared to a zero, the frame egresses the target port Untagged (i.e., the 4 byte DSA Tag will be removed from the frame). If the target port is a Provider Port (Section 4) see the NOTE below.
Trg_Dev	Target Device. These bits are used to define the target device's number. Use 0x00 for single chip switches (assuming the chip's DeviceNumber in Global 1 offset 0x1C = 0x00). From_CPU frames pass from chip to chip (using the switch port defined by the Device Mapping Table, Global 2 offset 0x06) until the frame finds a chip where the frame's Trg_Dev field matches the chip's DeviceNumber register.
Trg_Port	Target Port. These bits are used to define the target port's number (on the target device – see Trg_Dev above). Use 0x00 for Port 0, 0x01 for Port 1, 0x02 for Port 2, etc. From_CPU frames will Egress the Target Port on the Target Device (see above).
C	CFI bit. This bit is used as the frame's IEEE tag CFI bit if the frame egresses the target port Tagged (as defined by the Trg_Tagged bit above). If the target port is a Provider Port (Section 4) this bit will be the Provider Tag's DEI bit (see NOTE below).
PRI[2:0]	The frame's priority. PRI[2:1] are used to indicate which egress queue these frames are to be switched to unless overridden by the Priority Override Table (Global 2 offset 0x0F – Section 5.8). All three PRI bits are used as the frame's IEEE tag priority if the frame egresses the target port Tagged (as defined by the Trg_Tagged bit above).
VID[11:0]	The frame's VLAN identifier. These VID bits are ignored inside the switch on From_CPU frames. They are only used as the frame's IEEE VID if the frame egresses the target port Tagged (as defined by Trg_Tagged above).



Note

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

5.4 To_Sniffer DSA Tag

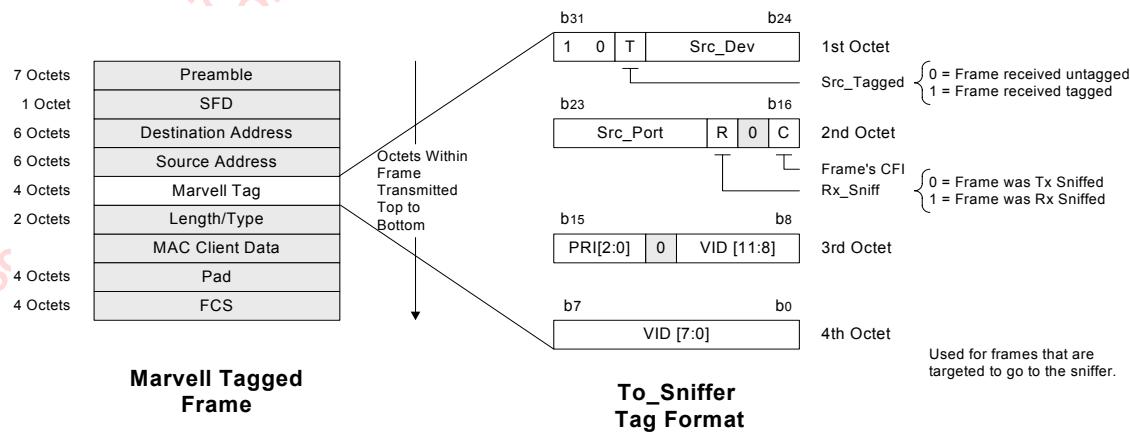
To_Sniffer DSA Tag frames are used to support chip-to-chip mirroring (Section 6.9). They are used to map Ingress Monitor Source (IMS) or Egress Monitor Source (EMS) frames to this device's Ingress Monitor Destination (IMD) or Egress Monitor Destination (EMD) port. Normally these frames do what they need to do without CPU processing. But if the CPU's port is the final IMD or EMD then the CPU will get these frames with the formats defined in Figure 31 and Table 16.

In an environment where more than one device is connected together using DSA Tag ports, the device(s) in the middle will receive To_Sniffer DSA Tag frames. These frames are sent through the intermediate switch unmodified. They egress from the port defined by the device's EgressMonitorDest register (Global 1 offset 0x1A) if the frame was Tx Sniffed (the R bit in Figure 31 = 0) or from the port defined by the device's IngressMonitorDest register (Global 1 offset 0x1A) if the frame was Rx Sniffed (the R bit in Figure 31 = 1).


Note

Provider Ports (Section 4) should not be the destination port for To_Sniffer frames, or any Mirror for that matter, since there is no way for the provider to distinguish these frames from the original frame data. On the other hand, a management CPU could be the destination for these frames because the DSA Tags will differentiate them from the normal frames.

Figure 31: To_Sniffer DSA Tag Format


Note

To_Sniffer DSA Tag frames are considered MGMT (management) frames. MGMT frames are processed differently inside the switch. See Section 5.6.

Table 16: From_CPU DSA Tag Fields

Frame's Field	Description
Src_Tagged	Source Tag Mode, i.e., the DSA Tag is placed in the frames on top of the standard IEEE Tag that was in the frame. Standard IEEE Tags contain an 0x8100 Ether type. This bit can only be set to a one if the frame came from a Normal Network port (Section 3) where 802.1Q is enabled (Section 3.2.2) and if the frame was IEEE Tagged. This bit will never be set to a one if the frame came in on a Provider Port (Section 4) nor if it entered a Normal Network port where 802.1Q Mode is Disabled even if the frame is IEEE Tagged (i.e., Port Based VLANs are being used). In these cases the DSA Tag is added to the frame leaving the rest of the frame's contents intact.
Src_Dev	Source Device. These bits are used to define the original source device's number where the frame first ingressed (i.e., the first device where the frame Ingressed from a Normal Network (Section 3) or Provider port (Section 4) before being switched to an Internal DSA Tag port). These bits come from the source device's DeviceNumber register (Global 1 offset 0x1C).
Src_Port	Source Port. These bits are used to define the original source port's number (on the source device above). 0x00 indicates Port 0, 0x01 indicates Port 1, 0x02 for Port 2, 0x03 for Port 3 etc. These bits always reflect the physical Src_Port of To_Sniffer frames even if the physical Src_Port is a Trunk port (Section 6.10).
Rx_Sniff	Receiver Sniff. This bit is set to a one if the frame came from an Ingress Monitor Source (IMS) port. This bit will be cleared to a zero if the frame came from an Egress Monitor Source (EMS) port. See Section 6.9 .
C	The original frame's CFI (Canonical Format Indicator) bit if the frame was IEEE tagged when it originally entered a Normal Network port (Section 3) on the switch. It is the original frame's DEI (Drop Eligible Indicator) bit if the frame was Provider tagged when it originally entered a Provider port (Section 4) on the switch.
PRI[2:0]	The frame's FPP priority as determined by the Ingress rules of the last devices that this frame entered (Section 3.4 for Normal Network ports and Section 4.2.2 for Provider ports).
VID[11:0]	The frame's VLAN identifier as determined by the Ingress rules of the last devices that this frame entered (Section 3.2.2.5 for Normal Network ports and Section 4.2.1 for Provider ports).



5.5

Cross-chip Features Using DSA Links

The switch fabric of one device is extended when two or more devices are linked together with Distributed Switch Architecture (DSA) ports. All the features of the device are supported cross-chip like Trunking ([Section 6.10](#)) and Mirroring ([Section 6.9](#)) along with Flow Control and VLANs (described below).

5.5.1

Cross-chip Flow Control

In this document, flow control refers to a generic method used to prevent frame loss at the expense of latency and QoS. When flow control is enabled on a port, “back pressure” is the specific mechanism used on half-duplex ports, while full-duplex ports use the IEEE “PAUSE” based mechanism. When flow control is enabled, the queue controller will use a different algorithm such that frames are not dropped when congestion occurs. When flow control is disabled, the queue controller will use a tail drop mechanism such that lower priority frames get dropped when congestion occurs ([Section 3.6](#)).

Flow control is supported cross-chip, without head of line blocking. The mixing of port flow control modes is also supported. Some ports in the switch can be flow controlled based (i.e., flow control is enabled – [Section 2.3.2](#), to [Section 2.3.4](#)), while others can be QoS based (i.e., flow control is disabled). When flow control enabled ports switch frames to other flow control enabled ports there will be no frame loss. This can occur at the same time that QoS enabled ports are switching frames to other QoS enabled ports. QoS actions will occur on these flows.

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

1. Enable cross-chip flow control messages, by setting FlowControlMessage to a one (Global 2 offset 0x05).
2. Ensure ForceFlowControlPri is set to a one and FC Pri is set to 0x7 (Global 2 offset 0x05).
3. Ensure the Priority Override Table (Global 2 offset 0x0F – [Section 5.8](#)) has entry 0x9 either disabled or enabled with a QPri setting of 0x3.
4. Enable flow control on both sides of each DSA link that (but don't enable flow control on the CPU's port). On DSA links Flow control must be forced on ([Section 2.3.4](#)).
5. Enable LimitOut with a value of 0xFF (Port offset 0x02 – [Section 2.3.5](#)) on at least one side of each DSA link.
6. Use the default values in the Flow Control Delays register (Global 2 offset 0x04).
7. Each device must have a unique DeviceNumber (Global 1 offset 0x01C) and the Device Mapping Table (Global 2 offset 0x06) must be properly configured in each chip indicating what port to egress frames targeted to a particular DeviceNumber.

5.5.2

Cross-chip 802.1Q VLANs

802.1Q VLANs are supported cross-chip. Each device supports the full 4,096 VID's in the VTU ([Section 8](#)), but each device only has storage for the port membership information for its own local ports. So when a VID is added with a defined port membership in one device, the same VID should be added to all devices in the switch defining the port membership for the local ports on each device (which could be the DSA port only). The DSA ports should be made members of all VIDs as they are extensions of the switch fabric and 802.1Q Secure and Check policy should not occur here (see [Section 5.7](#)). Either disable egress VID checking ([Section 3.8.3](#)) or ensure the DefaultVID for all ports (Port offset 0x07) is contained in the VTU. 802.1Q needs to be enabled on the DSA ports as well so that the VID lookups for port membership will occur.

5.5.3

Cross-chip Port Based VLANs

The device supports a very flexible cross-chip port based VLAN system that is used for all non-MGMT frames even if 802.1Q is enabled on the port. The 512 x 7 entry cross-chip Port VLAN Table (shown in [Figure 32](#)) is used for this feature. The table is accessed using two registers at Global 2 offsets 0x0B and 0x0C.

Cross-chip port based VLANs are supported on Forward frames received on Distributed Switch Architecture (DSA) or Ether type DSA¹ ports only (see Frame Mode in Port offset 0x04 and [Section 5.1](#)). It is not applied to any other frame type that enters the port.

Forward frames contain source port information about the frame in its Src_Dev (Source Device) and Src_Port/Src_Trunk (Source Port or Source Trunk) fields ([Section 5.1](#)). The Src_Is_Trunk field indicates the kind of data that is contained in the Src_Port/Src_Trunk field. This information in the DSA Forward frames indicates the Normal Network port or Provider port the frame originally entered when ingressing the collection of cascaded or stacked switch devices².

When a DSA Forward frame enters a DSA port³ the frame's source port information is examined and used to access the cross-chip Port VLAN Table (Global 2 offsets 0x0B and 0x0C). The data found in the table indicates which port or ports, in this device, the frame can egress. A one in a port's bit position indicates the frame is allowed to egress that port. Port 0's bit is in bit 0, Port 1's bit is in bit 1, etc. DSA ports that connect to other switch devices should always have their port's bit set to a one in the table for all table entries as the purpose of the table is to limit which Normal Network (or Provider) ports the frame can egress based upon what Normal Network (or Provider) port it originally entered the switch on.

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

- When 5 Bit Port is cleared to a zero (Global 2, offset 0x1D) the table is configured to be used with Marvell® Link Street® devices. Since all current Marvell Link Street® devices support no more than 11 ports per device and no more than 16 trunks per system, only the lower 4 bits of the Src_Port/Src_Trunk field is needed. In this mode the full 5 bits of the Src_Dev field is used. So the 9 bit Table Pointer is constructed as Src_Dev[4:0], Src_Port[3:0] (1st column of [Figure 32](#)).
- When 5 Bit Port is set to a one the table is configured to be used with Marvell DX or other devices where more than 16 ports or more than 16 trunks are used in the system. In this case the full 5 bits of the Src_Port/Src_Trunk field is needed. In this mode, the Src_Dev field is reduced to 4 bits. So the 9 bit Table Pointer is constructed as Src_Dev[3:0], Src_Port[4:0] (2nd column of [Figure 32](#)).

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

The Cross-chip Port VLAN Table is used in conjunction with the In-Chip Port VLAN Map ([Section 3.2.1.1](#)) and 802.1Q VLANs if enabled ([Section 3.2.2](#)). If any of these VLAN functions masks (or prevents) a frame from egressing a port that frame will not be allowed to go out that port.

MUX'ing frames to a Router like a Marvell DX device ([Section 6.8.1](#)) is supported with the Cross-chip Port VLAN Table. The In-Chip Port VLAN Map on the frame's original source port ([Section 3.2.1.1](#)) MUX'es the frame to the Router's port. If the Router 'returns' the frame back to this original source device⁴, the Cross-chip Port VLAN Table is then used to determine which ports the frame can egress.

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



Link Street®

88E6321/88E6320 Functional Specification

Figure 32: Cross-chip Port VLAN Table Formats

Src_Is_Trunk = 1 Src_Trunk = 0xF to 0x0	Pointer = 0x1FF	Src_Is_Trunk = 1 Src_Trunk = 0x1F to 0x00
Src_Dev = 0x1E Src_Port = 0xF to 0x0	Pointer = 0x1F0	Src_Dev = 0x0E Src_Port = 0x1F to 0x00
Src_Dev = 0x1D Src_Port = 0xF to 0x0	Pointer = 0x1E0	Src_Dev = 0x0D Src_Port = 0x1F to 0x00
Src_Dev = 0x1C Src_Port = 0xF to 0x0	Pointer = 0x1D0	Src_Dev = 0x0C Src_Port = 0x1F to 0x00
Src_Dev = 0x1B Src_Port = 0xF to 0x0	Pointer = 0x1C0	Src_Dev = 0x0B Src_Port = 0x1F to 0x00
Src_Dev = 0x1A Src_Port = 0xF to 0x0	Pointer = 0x1B0	Src_Dev = 0x0A Src_Port = 0x1F to 0x00
Src_Dev = 0x19 Src_Port = 0xF to 0x0	Pointer = 0x1A0	Src_Dev = 0x09 Src_Port = 0x1F to 0x00
Src_Dev = 0x18 Src_Port = 0xF to 0x0	Pointer = 0x190	Src_Dev = 0x08 Src_Port = 0x1F to 0x00
Src_Dev = 0x17 Src_Port = 0xF to 0x0	Pointer = 0x180	Src_Dev = 0x07 Src_Port = 0x1F to 0x00
Src_Dev = 0x16 Src_Port = 0xF to 0x0	Pointer = 0x170	Src_Dev = 0x06 Src_Port = 0x1F to 0x00
Src_Dev = 0x15 Src_Port = 0xF to 0x0	Pointer = 0x160	Src_Dev = 0x05 Src_Port = 0x1F to 0x00
Src_Dev = 0x14 Src_Port = 0xF to 0x0	Pointer = 0x150	Src_Dev = 0x04 Src_Port = 0x1F to 0x00
Src_Dev = 0x13 Src_Port = 0xF to 0x0	Pointer = 0x140	Src_Dev = 0x03 Src_Port = 0x1F to 0x00
Src_Dev = 0x12 Src_Port = 0xF to 0x0	Pointer = 0x130	Src_Dev = 0x02 Src_Port = 0x1F to 0x00
Src_Dev = 0x11 Src_Port = 0xF to 0x0	Pointer = 0x120	Src_Dev = 0x01 Src_Port = 0x1F to 0x00
Src_Dev = 0x10 Src_Port = 0xF to 0x0	Pointer = 0x110	Src_Dev = 0x00 Src_Port = 0x1F to 0x00
Src_Dev = 0x0F Src_Port = 0xF to 0x0	Pointer = 0x100	
Src_Port = 0xF to 0x0 Src_Dev = 0x0E	Pointer = 0x0F0	
Src_Port = 0xF to 0x0 Src_Dev = 0x0D	Pointer = 0x0E0	
Src_Port = 0xF to 0x0 Src_Dev = 0x0C	Pointer = 0x0D0	
Src_Port = 0xF to 0x0 Src_Dev = 0x0B	Pointer = 0x0C0	
Src_Port = 0xF to 0x0 Src_Dev = 0x0A	Pointer = 0x0B0	
Src_Port = 0xF to 0x0 Src_Dev = 0x09	Pointer = 0x0A0	
Src_Port = 0xF to 0x0 Src_Dev = 0x08	Pointer = 0x090	
Src_Port = 0xF to 0x0 Src_Dev = 0x07	Pointer = 0x080	
Src_Port = 0xF to 0x0 Src_Dev = 0x06	Pointer = 0x070	
Src_Port = 0xF to 0x0 Src_Dev = 0x05	Pointer = 0x060	
Src_Port = 0xF to 0x0 Src_Dev = 0x04	Pointer = 0x050	
Src_Port = 0xF to 0x0 Src_Dev = 0x03	Pointer = 0x040	
Src_Port = 0xF to 0x0 Src_Dev = 0x02	Pointer = 0x030	
Src_Port = 0xF to 0x0 Src_Dev = 0x01	Pointer = 0x020	
Src_Port = 0xF to 0x0 Src_Dev = 0x00	Pointer = 0x010	
Src_Port = 0xF to 0x0	Pointer = 0x000	

Cross Chip Port VLAN Table
when 5 Bit Port = 0Cross Chip Port VLAN Table
when 5 Bit Port = 1

5.6

Switch Handling of DSA MGMT Frames

As stated in [Section 5.1](#), Forward DSA Tag frames are processed as if they entered a Normal Network port ([Section 2](#) and [Section 3](#)). But the other three major DSA Tag types, To_CPU ([Section 5.2](#)), From_CPU ([Section 5.3](#)) and To_Sniffer ([Section 5.4](#)), are all considered MGMT (management) frames and they are processed differently as follows:

- The various frame types get mapped to their destination ports as defined in there respective sections. No other frame mapping functions occur. DA mapping, VID mapping, Port Based VLAN mapping, Trunk Load Balancing, Layer 2 Policy, etc. are all ignored. It does not matter if the ingress or egress port is Blocked ([Section 3.1.1](#)), these frames will go through.
- QoS priority overrides are ignored except for the Priority Override Table (Global 2 offset 0x0F) which is used for Secure Control Technology ([Section 5.8](#)).
- SA Filtering is ignored except for Drop On Unlock, i.e., frames from a potential hacker - see Note ([Section 3.1.2.3](#)).
- 802.1Q Secure and Check frame drop conditions are ignored ([Section 3.2.2.8](#)).
- Discard Untagged and Discard Tagged frame drop conditions are ignored ([Section 3.2.2.3](#) and [Section 3.2.2.4](#)).
- Source addresses are not learned or refreshed ([Section 2.4.6](#)).



Note

DropOnUnlock ([Section 3.1.2.3](#)) will discard MGMT frames based on SA. This feature is not intended for DSA Ports. Instead it is intended for Normal Network or Provider Ports where frames can become MGMT as soon as they enter the switch port based on the frame's DA. Supporting DropOnUnlock in this case can help prevent BPDU (or any MGMT) DoS Attacks by discarding all MGMT frames from a particular source (or sources).



5.7

Proper Usage of DSA Tag Ports

Distributed Switch Architecture (DSA) Tag ports are used to interconnect devices to each other and to the management CPU. These internal ports are an extension of the switch fabric and therefore they are inherently trusted. DSA Tag ports typically carry frames from many different source ports (i.e., frames from potentially non-trusted external ports set to either Normal Network mode - [Section 3](#), or Provider mode - [Section 4](#)).

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

- Policy ([Section 3](#)) needs to be done on the external ports (Normal Network or Provider) as the frames first enter the switch. This includes Denial of Service (Dos) attack prevention using PIRL resources ([Section 3.5](#)) on all frames that go to the CPU (MGMT, ARPs, IGMP, etc.).
- DSA Tag ports need to be an open pipe of data into the switch device. Therefore Policy must not be done on DSA Tag ports as that policy will affect multiple source and destination ports. This means that none of the feature described in [Section 3](#) should be enabled on DSA Tag ports with the exception of VLAN mapping ([Section 3.2.2.8](#)).
- Basic switch operations ([Section 2](#)) with the exception of 802.1X ([Section 2.4.7](#)) must be done on DSA Tag ports so that learning and switching can occur in the local device. 802.1X should be done on the external ports only.
- Learn2All should be enabled for Cross-chip learning (Global 1 offset 0x0A). The MessagePort bit (Port offset 0x05) must be set on all DSA links (except for the CPUs link) for this to work.
- VLAN switch operations (the mapping portions of [Section 3.2](#), not the Policy, i.e., frame dropping portions) must be done on DSA Tag ports so the local VLAN isolation can be done.
 - Use In-chip Port Based VLANs to prevent loops ([Section 3.2.1.1](#)).
 - Use Cross-chip Port Base VLANs ([Section 5.5.3](#)) when at least one external port is in a Port Based VLAN mode. All switch devices should have their Port VLAN Table filled for every external (source) port configured in this mode.
 - Use 802.1Q VLAN MemberTag mapping and tagging ([Section 3.2.2](#)) when at least one external port is using any of the three 802.1Q enabled modes. All switch devices should have every VID being used in the switch defined and loaded into their local VTU.
- Enable Cross-chip flow control if needed ([Section 5.5.1](#)).
- Force the link up on all (G)MII DSA ports, including the CPU's port ([Section 2.3.7](#) – Link will not come up automatically on internal ports).
- Mirroring (all types) can only be done on DSA Ports that are connected directly to a CPU. Chip-to-chip interconnected DSA ports must not enable any mirroring.
- Frame type priority overrides can be done for Secure Control Technology only ([Section 5.8](#)).
- Do not use the Ether type DSA frame ([Section 5.9](#)) format on chip-to-chip interconnections. It is designed to be used on the port directly connected to a CPU.
- Properly configure each device's Device Mapping Table (Global 2 offset 0x06 – [Section 5.3](#)).

5.8**Secure Control Technology (SCT)**

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

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

The Priority Override Table (Global 2 offset 0x0F) is used for SCT. In switch devices that are not connected directly to the CPU, the table is set to QPri 3 for all MGMT except mirrors. In the switch device connected directly to the CPU, the table is set to the desired priority depending upon the MGMT type. [Table 17](#) shows an example.

Table 17: Secure Control Technology Example

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

5.9

Ether Type DSA Tag

An alternate DSA Tag mode, called Ether type DSA Tag is supported and defined in [Figure 33](#) and [Table 18](#). Ether type DSA encapsulates the standard DSA Tags, described above, after a programmable Ether type. The Ether type DSA mode is optimized for switch to CPU interconnections for the following reasons. Ports in this mode will:

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



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

The proper usage is to set the device port used to connect to the management CPU to Ether Type DSA Tag mode (i.e., FrameMode = 0x3 at Port offset 0x04) and set the ports used to interconnect devices to each other in regular DSA Tag mode (i.e., FrameMode = 0x1) while the external ports are set to either Normal Network mode ([Section 3](#)) or Provider mode ([Section 4](#)). All FrameMode port types can co-exist as the same time on different ports of device. The format of the frames will be converted as needed.

Figure 33: Ether Type DSA Tag Format

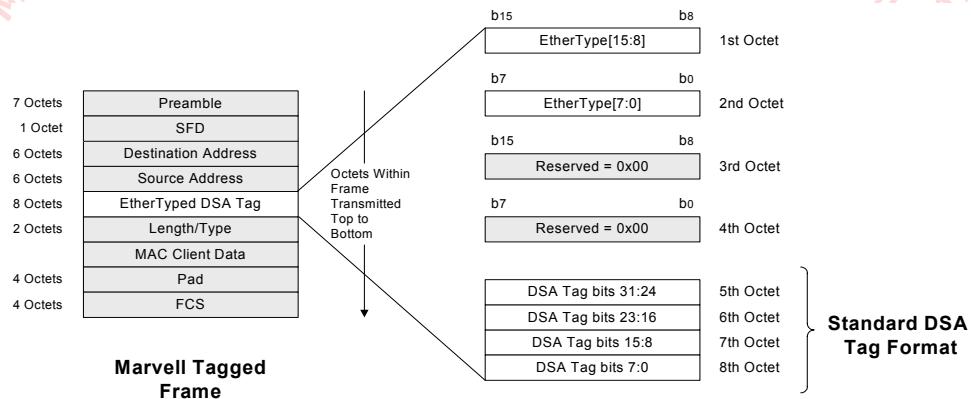


Table 18: Ether Type DSA Tag Fields

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

6

Advanced Switch Functions

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

This section covers the following topics:

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

6.1

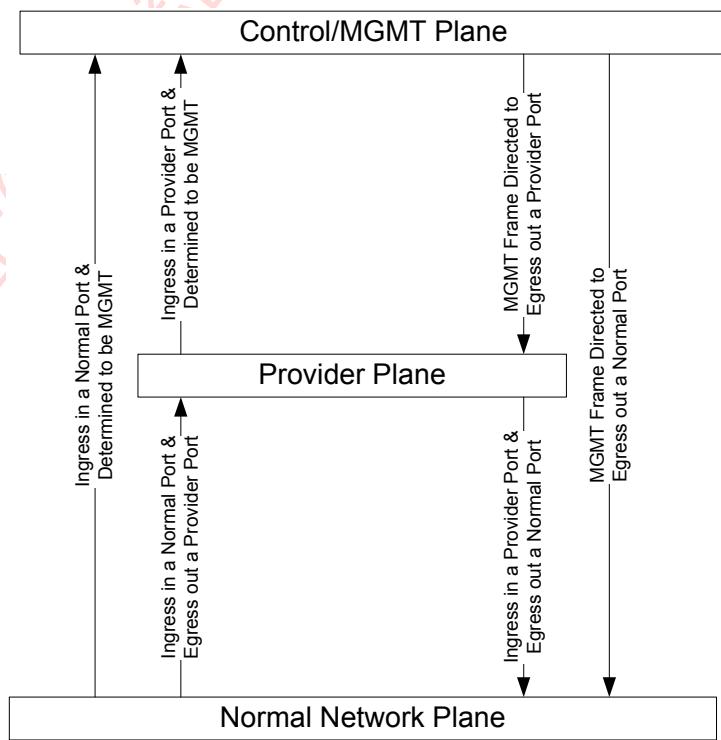
Management Frames to and from the CPU

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

The device supports a concept of a MGMT (management) data plane where control and switch management frames travel under different rules from Normal data (otherwise called Normal Network – [Section 3](#), and/or Provider data – [Section 4](#)). MGMT frames can and do use the same physical ports as Normal frames but they are processed differently.

[Figure 34](#) shows the concept of these separate planes and the linkage between them. The following sections cover the linkage (i.e., how to get some Normal frames to the MGMT Plane and back again) and how MGMT frames are processed inside the switch.

Figure 34: Normal and Provider vs. Control Data Planes



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

6.2

Spanning Tree Support

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

- Detection of BPDU¹ frames entering Network (or provider) (external switch) ports. These frames are called MGMT (management) frames in the device. They are detected by loading the BPDU's multicast address (01:80:C2:00:00:00) into the address database with a MGMT EntryState indicator ([Section 7.3.1](#)) or by using the Rsvd2Cpu bits in MGMT Enable register (Global 2 offset 0x03).
- Tunneling of BPDU frames through Blocked ports. Blocked ports are controlled by the Port's PortState bits (see [Section 3.1.1](#) and [Section 3.2.3](#)). If a port is in the Blocked state, all frames are discarded except for frames with a DA address that is considered a MGMT address as defined in the step above.
- Redirection of BPDU frames. BPDU frames that enter a Network (or provider) port need to go to the CPU only, even though they are multicast frames. This task is handled in the BPDU frame detection phase above by mapping the BPDU's multicast address to the CPU port directly or to the port that is to be used to cascade these frames to the CPU (this is set with the value of the DPV bits when the address is loaded) or by the device's CPUDest register (Global 1 offset 0x1A). Cascaded BPDU frames egress the 1st device with a To_CPU DSA Tag ([Section 5.2](#)).
- Cascading of BPDU frames. BPDU frames that enter a DSA Tag port must enter it with a To_CPU DSA Tag. These frames are mapped directly, without modification to the device's CPUDest (Global 1 offset 0x1A). The CPUDest registers are used to form a path from all the Network ports in the switch to the CPU.
- Source Port information. The CPU needs to know the physical source port of the BPDU frame. This information is supplied in the frame's To_CPU DSA Tag (see [Section 5.2](#)) that is sent to the CPU.
- CPU transmission of BPDU frames. The CPU needs to be able to transmit BPDU frames out any physical port of the switch. This control is supported with the From_CPU DSA Tag data that the CPU needs to put on these frames before they are transmitted into the switch ([Section 5.3](#)). The Device Mapping table (Global Control 2 offset 0x06) is used to map From_CPU frames that are not destined for a local device (as marked in the DSA Tag's Trg_Dev field) out the correct port to get it to the next device. If the next device that receives the From_CPU frame is not the final destination, then process repeats until the frame gets to the target device. Once there, it will send the From_CPU frames out the target port (as marked in the DSA Tag's Trg_Port field) with the DSA Tag removed from the frame assuming the target port was a Network port.

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



Note

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

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



6.3

Ingress MGMT/BPDU Frame Detection

The device supports two methods of management/802.1D BPDU frame detection and mapping for ingressing frames from Normal Network and/or Provider ports. These two mechanisms support IEEE industry standards like STP¹, LAC², and OAM³, as well as any company proprietary protocol. Both of these mechanisms:

- Detect that a frame is special by examining the frame's DA
- Assign these special frames to a category called MGMT (for management)
- Allow MGMT frames and only MGMT frames to ingress and egress Blocked ports (see Port States in [Section 3.1.1](#))
- Set the priority on these MGMT frames overriding all other QoS decisions on the frame
- Map these MGMT frames to a port where a management CPU is directly or indirectly connected

6.3.1

Reserved Multicast Address Support

The first mechanism for MGMT frame detection is optimized to support 802.1D's 16 reserved multicast addresses. Any or all of the 16 multicast addresses in the range of 01:80:C2:00:00:0x⁴ can be treated as MGMT addresses in the device. The Rsvd2Cpu register bits in the MGMT Enables 0x register (Global 2 offset 0x03) determines which of these 16 addresses are treated as MGMT and which are not as long as the Rsvd2Cpu bit in the Switch Management register (Global 2 offset 0x05) is also set to a one.

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

Any frame, regardless of its VLAN identifier (VID) or FID⁵ assigned to it, whose DA matches an enabled reserved multicast address will be considered a MGMT frame. It will be given the priority defined in the MGMT_Pri bits (Global 2 offset 0x05) and mapped to the port defined by the global CPUDest register (Global 1 offset 0x1A).

6.3.2

New and Proprietary Protocol Support

The second mechanism for MGMT frame detection is optimized to support any new, or yet to be defined, standard and/or proprietary DA based protocol. It can also be used to map any of the 32 reserved multicast addresses defined above, where the VID of the frame must be considered in the MGMT determination⁶. Any address, multicast or unicast, can be treated as a MGMT address in the device in this way. The Address Translation Unit (ATU) is used in this case. The required MAC address must be loaded into the ATU with a MGMT EntryState value, the required priority for the frame and where the frame is to be mapped (see [Section 7.3.1](#)).

Any frame whose DA matches an ATU entry with a MGMT EntryState will be considered a MGMT frame. It will be given the priority defined in the ATU's entry and mapped⁷ to the port or ports defined by the entry's Destination Port Vector (DPV). The ATU supports multiple address databases ([Section 2.4.8](#)) so the DA must appear in the Forwarding Information Database (FID) assigned to the frame for the frame to be considered a MGMT frame. This feature allows a DA to be considered a

1. Spanning Tree Protocol
2. Link Aggregation Control
3. Operational, Administration, and Maintenance
4. Frames with a DA of 01:80:C2:00:00:01 are always treated as Pause frames and are discarded and never mapped.
5. FID is a Forwarding Information database number assigned to each frame to support multiple address databases (see [Section 2.4.8](#)).
6. If the second mechanism is being used to map any of the 32 reserved multicast addresses the bit that corresponds to the required address in the Rsvd2CpuEnables must be cleared to a zero since the first mechanism takes priority over the second mechanism.
7. EntryState = 0xE should be used so the ATU entry can force the MGMT FPri to 0x7 and QPri to 0x3 by setting its P bits to 0x7.

MGMT address in some address databases and not in others. But it also requires that the DA be loaded multiple times, once for each address database that needs to use this address as a MGMT address.



Note

Frames that are considered MGMT by their DA are considered MGMT in the Ingress section. That means that the frame cannot be filtered or have its priority modified due to any Normal Network Policy ([Section 3](#)) with the exception of Ingress Rate Limiting of MGMT frames ([Section 3.5](#)) or DropOnUnlock ([Section 3.1.2.3](#)) which will discard MGMT frames based on SA. The DropOnUnlock feature is not intended for DSA Ports. Instead it is intended for Normal Network or Provider Ports where frames can become MGMT as soon as they enter the switch port based on the frame's DA. Supporting DropOnUnlock in this case can help prevent BPDU (or any MGMT) DoS Attacks by discarding all MGMT frames from a particular source (or sources).

6.4

Other Ingress MGMT Frame Detection

Most 802.1 protocols require special frames to get to the management CPU. These are handled by looking at the frame's Destination Address (see [Section 6.3](#)). There are other ways to move a Normal Network (or Provider) Plane frame up to the MGMT Plane (as shown in [Figure 34](#)). These are:

- ARP Mirror frames ([Section 3.3.3](#))
- IGMP/MLD Snooped (or Trapped) frames ([Section 3.3.4](#)).
- Layer 2 Policy Traps or Policy Mirrors ([Section 3.1.3](#))



Note

Frames that are considered MGMT by other than the frame's DA are not considered MGMT in the Ingress section (by they are considered MGMT in egress and then on until they reach their destination). This means that these frames can be discarded due to any Normal Network Policy ([Section 3](#)). This ensures that ARPs, IGMP frames, etc., are members of a port's VLAN before they will be sent to the CPU as MGMT.

6.5

MGMT Frames to Normal or Provider Egress

[Figure 34](#) shows MGMT frames going back down to the Normal Network Plane or to the Provider Plane. This is because 802.1 protocols require that the CPU transmit special frames out the ports. If the target egress port is a Normal Network port ([Section 3](#)) or a Provider port ([Section 4](#)) then the frames need to look like normal IEEE frames.

The CPU uses From_CPU DSA Tag frames ([Section 5.3](#)) to get the special frames out a specific port. The egress logic will automatically convert the From_CPU DSA Tag to a normal IEEE frame format if the egress port is configured in Normal Network or Provider mode (FrameMode, Port offset 0x04).



Note

From_CPU DSA Tag frames are considered MGMT by the switch until the frame egresses the target port, i.e., they will egress Blocked ports ([Section 3.1.1](#)).



Link Street®

88E6321/88E6320 Functional Specification

6.6

Proper Connection to a Management CPU

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

Desired DA's, like the CPU's own MAC address, can be loaded into the ATU as static ([Section 7.3.3](#) – the CPU's port will have to be a member of the frame's VLAN too, unless the CPU's MAC address is loaded as MGMT, but then it will tunnel through Blocked ports). ARP's can be mirrored ([Section 2.3.3](#)) and IGMP/MLD frames can be trapped ([Section 2.3.4](#)). Other 802.1 protocol frames can be gotten to the CPU based on the frame's DA ([Section 6.3](#)).

Other barrier options to isolate the CPU are to use Port Based VLANs ([Section 3.2.1](#)) or 802.1Q VLANs ([Section 3.2.2](#) – where the DefaultVID on all ports, Port offset 0x07, is defined in the VTU).

The CPU can be protected by Egress Rate Shaping using frame's per second ([Section 3.8.5.1](#)). Ingress Rate Limiting ([Section 3.5](#)) on all the Normal Network and Provider Ports to limit the number of MGMT frames and/or ARP frames that are allowed to enter the switch can also be used to protect the CPU.

6.7

Proper Connection to a Router

Routers can be a CPU or some other switching device connected to a port that performs routing or higher layer switching. In this case the router needs to get all the frames so that they can be processed. ‘Barriers’ between some ports need to be set up to ensure frames get processed before egressing other ports on the switch. While at the same time normal switching may be allowed between some ports on the device depending upon which LAN each port belongs to. [Section 3.2.1.2](#) gives a couple of examples on how to set up these ‘barriers’ using Port Based VLANs.

Usage of the Address Database needs to be considered. It may be advantageous to disable learning on some ports (LearnDisable in Port offset 0x06) where switching never occurs (like the WAN to CPU and the CPU to WAN paths). Other situations may need multiple and separate Address Databases ([Section 2.4.8](#)) and keep learning enabled on the ports. Forcing the frames to the Router may need to be done by ignoring the results of the Destination Address search ([Section 6.8](#)).

The performance of CPU routing can be greatly increased by using the Marvell® Header on the CPU's port because it aligns the IP portion of the Ethernet frame to 32-bit boundaries in the CPU's memory. The Header is also needed to limit where frames go (for single chip switch systems – in multi chip systems the same effect may be accomplished by using Cross-chip Port Based VLANs – [Section 5.5.3](#) where the CPU can limit where frames go based upon the source port information it puts in the Forward DSA Tag).

6.7.1

Switch Ingress Header for Routers

The CPU in a router needs to perform many functions. One of those functions is to route IP frames from a WAN to or from LAN ports and another is to bridge frames between one VLAN and another VLAN. The devices Ingress Header mode increases the performance of both of these functions. Any port can be configured to support an Ingress Header by setting the Header bit¹ in the port's Port Control register (Offset 0x04) but only the port directly connected to a CPU should be configured in this way.

The Ingress Header accelerates the CPU's performance when routing IP frames by aligning the IP portion of the frame to 32-bit boundaries. This is accomplished by prepending the frame with two extra bytes of data. Bridging between VLAN ports sometimes requires the switch to support multiple address databases (one for each VLAN) so that the same MAC address can be used on multiple

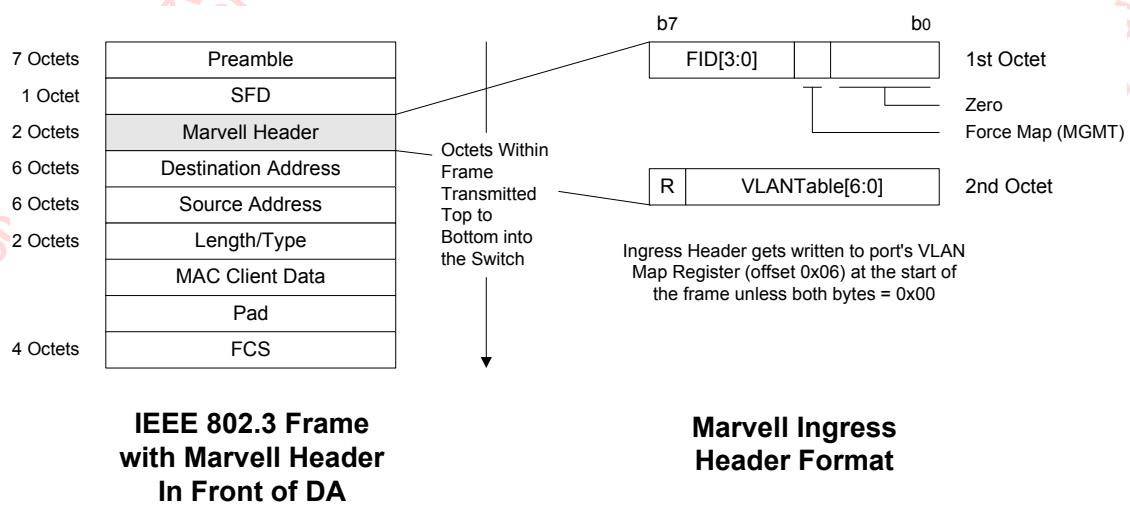
1. The Header bit enables the Marvell Header mode for both ingress and egress.

VLANs. Since the CPU is generally a member of all VLANs, it must inform the switch which VLAN to use on a given frame (and thus which address database to use). This is accomplished by using an Ingress Header with a non-zero value as defined in [Figure 35¹](#). When the Ingress Header is seen with a non-zero value its contents are written to the port's Port Based VLAN Map register (Offset 0x06) prior to the start of the rest of the frame. The frame is then processed by the switch using this new information. In this way, the CPU can direct which port based VLAN and address database to use on every frame at wire speed.

When the Ingress Header mode is enabled on a port, the first two bytes of the frame (just before the DA) are used to control the switch. The Ingress Policy block removes the Header from the frame, causing the frame to be two bytes smaller in size, and overwrites the frame's FCS with a new FCS. This adjustment makes the frame normal for the rest of the network since the Header's data is intended for the switch only. Frame size checking is performed on the adjusted frame size. This means the CPU must always add two bytes of data to the beginning of every frame it sends into the switch (if the Header mode is enabled).

The Ingress Header gives the CPU the ability to control which VLAN (port based), learning mode and address database to use on the frame that it just received. It may not always be convenient to use the CPU to manipulate the appropriate switch registers. If the CPU directs the switch to process the frame based upon the switch's current Ingress policy then the CPU sets the Header data in the frame to all zeros (i.e., it prepends the frame with two extra bytes of zeros). This zero padding indicates that the switch should ignore the Header's data and process the frame normally (after the Header's data is removed from the frame).

Figure 35: Ingress Header Format



When an Ingress Header contains a non-zero value, its contents are written directly to the port's Port Based VLAN Map Register.



The Marvell® Header can only modify the lower 4-bits of the port's FID. The upper 2-bits come from the port's FID[5:4] register bits from Port based VLAN Map register - Offset 0x06.

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

6.7.2

Switch Egress Header

If a CPU wants to have the IP frame data of the Ethernet frame aligned to 32-bit boundaries for faster routing, the devices support a Marvell® Header Mode that inserts two bytes into the frame just before the frame's Destination Address. Any port can be configured this way by setting the Header bit in the port's Port Control register (Offset 0x04) but only the CPU's port should be configured this way¹.

When the Egress Header mode is enabled on a port, two extra bytes are added to the beginning of the frame just before the frame's DA and a new CRC is calculated for the frame. When the frame is received by the CPU the Header will be the first two bytes of the frame in memory. If the CPU's MAC needs to process the frame for filtering or for other reasons, the MAC must be aware that the frame data has been shifted down by two bytes.

The Egress Header can be configured using the Header Type bits in the Global Control 2 Register (Offset 0x1C). The format of the two types of Egress Header are described below:

Figure 36: Egress Header Format - Header Type = 00 (Original Header)

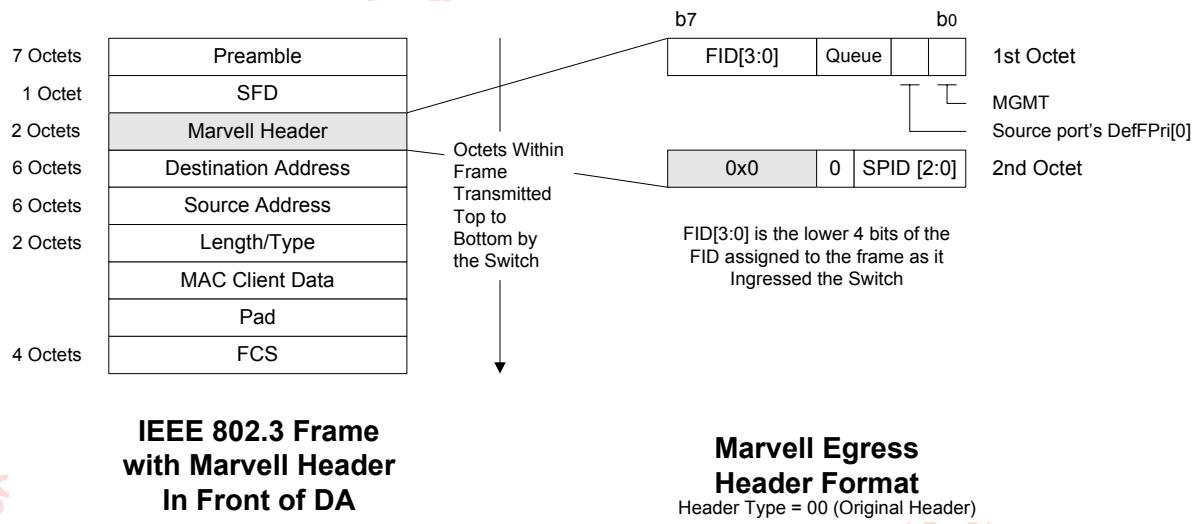


Table 19: Egress Header Fields - Header Type = 00 (Original Header)

Field	Description
FID[3:0]	Forward Information Database Number. This field represents the lower 4 bits of the address database number assigned to this frame when it Ingressed into this device. It can be used to indicate the logical port based VLAN number of the source port.
Queue	The frame's queue priority. This is the queue priority of the frame that is being transmitted out. This enables switches to communicate queue priority with their link partner.
DefFPRI[2:0]	The frame's source port's Default Frame Priority bit 0. This value is set in the register at Port offset 0x07 of the frame's source port.
MGMT	Management. When this bit is set to a one it indicates the frame is a Management frame as determined by any of the methods defined in Section 6.3 and Section 6.4 .
SPID[2:0]	The Source Port ID. These bits indicate at which physical port the frame entered this device (i.e., it is assigned by the last physical device this frame entered). An SPID of all zeros indicates Port 0. An SPID of 0x1 indicates Port 1. 0x2 indicates Port 2, etc.

1. The Header bit enables the Header mode for both Ingress and Egress

Figure 37: Egress Header Format - Header Type = 01 (Single chip MGMT Header)

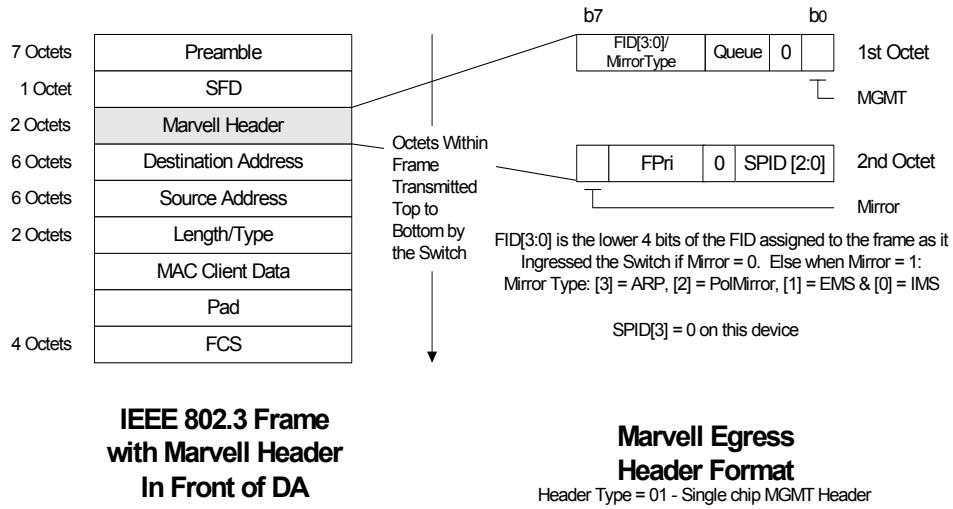


Table 20: Egress Header Fields - Header Type = 01 (Single chip MGMT Header)

Field	Description
FID[3:0]/MirrorType	Forward Information Database Number or Mirror Type bits. If the Mirror bit below is cleared to zero these bits contain the Forward Information Database Number. This field represents the lower 4 bits of the address database number assigned to the frame when it Ingressed into this device. It can be used to indicate the logical port based VLAN number of the source port. When the Mirror bit below is set to a one these bits contain the Mirror Type information where bit [3] is set to a one for ARP frames, bit [2] for PolMirror frames, bit [1] for Egress Monitor Source frames and bit [0] for Ingress Monitor Source frames. More than one Mirror Type bit can be set to a one for a given frame.
Queue	The frame's queue priority. This is the queue priority of the frame that is being transmitted out. This enables switches to communicate queue priority with their link partner.
MGMT	Management. When this bit is set to a one it indicates the frame is a Management frame as determined by any of the methods defined in Section 6.3 and Section 6.4 .
Mirror	Mirrored frame. When this bit is set to a one it indicates that this frame was mirrored to this port and the FID[3:0]/MirrorType bits indicate the reason for the mirror.
FPri	The frame's frame priority. This is the frame priority of the frame that is being transmitted out as it was assigned to the frame.
SPID[2:0]	The Source Port ID. These bits indicate at which physical port the frame entered this device (i.e., it is assigned by the last physical device this frame entered). An SPID of all zeros indicates Port 0. An SPID of 0x1 indicates Port 1. 0x2 indicates Port 2, etc.



Link Street®

88E6321/88E6320 Functional Specification

6.8

MUX'ing or Ignoring Address Translation

The device supports the ability to ignore the results of a frame's DA lookup in the Address Translation Unit. The use of the DA mapping results can be enabled or disabled on a port-by-port basis by changing the value of the port's MapDA bit (Port offset 0x08). DA lookups always take place, regardless of the setting of port's MapDA bit, however. This is done so that DA lookups that are found to be MGMT entries in the ATU ([Section 6.3.2](#)) are always mapped (even if the MapDA bit is configured to ignore the results). The application of all other switch policies are not effected by the port's MapDA bit.

When DA mapping is disabled (the port's MapDA bit equals zero) all non-MGMT frames that enter the port will be mapped based on the VLAN rules that are applied to the frame ([Section 3.2](#)) along with the Egress Flooding rules ([Section 3.8.1](#) and [Section 3.8.2](#)). These bits can be configured in such a way that all non-MGMT frames that enter a port egress out a specific port (even the frame's source port, [Section 3.3.1](#)). This can be used for the following applications:

6.8.1

Passing Frames to a Router

The MapDA bit can be used to MUX all non-MGMT frames that enter a port to go out another port where a router can perform more processing on the frame. VLANs are used to get the frame to the router's port ([Section 3.2](#)). MGMT frames will be mapped as they normally would ([Section 6.3](#)).

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

Alternatively, use Distributed Switch Architecture (DSA) Tags on the port connected to the router ([Section 5](#)) and enable the global LoopbackFilter bit (Global 2 offset 0x05). DSA Tags contain the original source port information in the frame. So when the router sends the frame back into the switch (with the DSA Tag portion of the frame unmodified) the switch knows to prevent sending the frame back out its original source port (when LoopbackFilter is set to a one).

6.8.2

Operational, Administration, and Maintenance (OAM) Loopback

The MapDA bit can be used to MUX all non-MGMT (i.e., non-OAM) frames that enter a port to go back out the port they came in on. Port Based VLANs are used to get the frame to go back out the original port and only the original port ([Section 3.3.1](#)). MGMT frames, including OAM control frames, will be mapped as they normally would ([Section 6.3](#)).

**Note**

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

6.9

Port Monitoring Support

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

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

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

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

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

Cross-chip port monitoring requires the following:

- Any time a monitor source is enabled in a device, the associated monitor destination must also be defined.
- A monitor source cannot be a DSA Tag port unless it is the CPU's port. Basically it is best if the monitor source ports are Normal Network ports and final monitor destination ports are also Normal Network ports.
- If the final monitor destination is cross-chip in another device, the monitor destinations in each device must form a complete path toward the final monitor destination port using DSA Tag enabled ports on both sides of the connections.
- All final Network monitor destination ports (both EMD and EMD) must be isolated from all the other local ports in the switch to prevent these ports from getting flooding frames from non-monitor source ports. The best way to isolate these ports is to use port based VLANs ([Section 3.2.1](#)).
- The final egress monitor destination port's 802.1Q Mode and VID MemberTag information must be configured to match the egress monitor's source port's configuration. If this is not done, the frame will still egress the final EMD port with the correct data in the frame but its tag mode (i.e., egressing tagged or egressing untagged) may not match the way the frame egressed the original EMS port.

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

- Frames received by the MAC but were not stored by the switch due to a lack of memory



Link Street®

88E6321/88E6320 Functional Specification

- Frames received with a bad CRC (unless the CRC is fixed by the ForceFCS bit being set on the port, or if the bad CRC is ignored by the Allow Bad bit being set on the port - Port offset 0x08)
- Pause frames that are received
- Frames with a size less than 64 bytes or greater than the maximum size allowed
- Frames discarded due to Ingress Rate Limiting ([Section 3.5](#))
- Frames received but sampled¹ due to Ingress Rate Limiting ([Section 3.5](#))

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

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

1. One of the port's Ingress Rate Limiting resources can be used to mirror 1 of N received IMS frames from the port (see [Section 3.5](#)).

6.10

Port Trunking Support

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

6.10.1

Trunk Address Learning

When a frame enters a Trunk Port its Source Address (SA) is learned (or loaded) with its association being to the ingress port's TrunkID number (its T bit will be a one - [Section 7.3.1](#)). This way the contents of the address database contain the same association with the frame's SA regardless of what link of the trunk the frame entered the switch. If this frame egresses a DSA Port ([Section 5](#)) it will be marked as coming from a trunk port and its Src_Port/Src_Trunk field will contain the TrunkID of the first trunk port the frame entered. This ensures that all devices in the switch learn this frame's SA with the source port's TrunkID.

6.10.2

Trunk Address Searching

When frames are destined back toward a trunk the frame will have its Destination Address (DA) searched for in the address database. If the frame's DA is unknown the frame will try to flood out all ports of the trunk (this is OK so far as this will be fixed with load balancing). If the frame's DA is found the entry will indicate the entry is mapped to a trunk (its T bit will be a one - [Section 7.3.1](#)) and the entry's DPV bits will contain the TrunkID associated with the frame's DA. This TrunkID needs to be converted into a DPV (Destination Port Vector) the rest of the switch can use. This is accomplished by accessing the Trunk Mapping table (Global 2 offset 0x08) using the TrunkID that was in the ATU's entry.

6.10.3

Trunk Mapping

The Trunk Mapping table (Global 2 offset 0x08) needs to be configured in each device in the switch by software for each TrunkID number that is in use. It is used to convert found DA searches that return an entry that is associated with a TrunkID (one with its T bit set to a one - [Section 7.3.1](#)) into a DPV (Destination Port Vector). Software configures each TrunkID entry in the table by placing a one in the entry for every port on the local device that is a direct or indirect member of the Trunk. Direct members of a trunk are ports with their Trunk Port bit set to a one (Port offset 0x05) with a TrunkID (also at Port offset 0x05) value equal to the Trunk Mapping table's entry being modified. Indirect members of a trunk are local device DSA Ports ([Section 5](#)) that connect to another switch device that contains direct members of the same TrunkID. The multiple bits being set in the resulting DPV ensures the frame is mapped to all possible members of the trunk (this is OK so far as this will be fixed with load balancing).

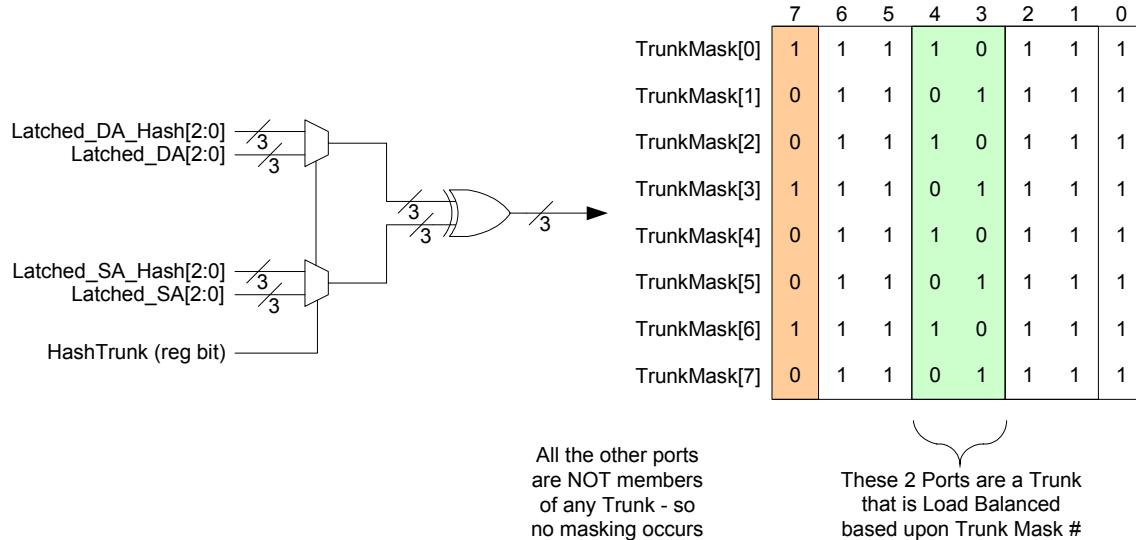
6.10.4

Load Balancing

Load balancing is used to ensure frames only egress one link (or port) member of a trunk. DA/SA based load balancing is used by configuring the Trunk Mask table (Global 2 offset 0x07). Software must configure all eight entries for all ports in any device that contains direct members of a trunk ([Section 6.10.3](#)). Load balancing will not occur on MGMT frames ([Section 6.1](#)).

The Trunk Mask table example ([Figure 38](#)) contains a bit per port for eight mask or load balance settings. Each frame that ingresses the switch selects one of the eight possible masks (depending upon the frame's DA and SA) and the selected mask is used to ensure each frame egresses only one port on each trunk.

Figure 38: Trunk Mask Load Balancing Table - Example



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

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

Figure 38 shows an example where ports 3 and 4 form one trunk. Ports that are not members of any trunk must have their Trunk Mask bits set to a one in all Trunk Mask entries. This keeps normal data flowing out those ports regardless of the frame's DA and SA.

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

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

6.11

Precise Time Protocol (PTP)

The IEEE 802 standards body has thus far defined ethernet as asynchronous in nature where end stations don't operate off of a common time base and neither do they have a concept of time. There are several applications that would benefit from incorporating the concept of time, making an isochronous Ethernet. Some of these applications are media streaming applications like IPTV, high definition audio video consumer appliance traffic, telecom networks, industrial automation, etc.

The time and/or clock information is propagated through a network using a protocol defined in the IEEE802.1AS standards committee. This standard is called Precise Timing Protocol (PTP).The PTP achieves the following by exchanging control packets periodically:

- Elects one of the network elements which has a best quality clock as the Grand Master for the PTP network. All the non-Grand Master nodes become PTP slave nodes.
- PTP slave nodes derive their frequency and time-of-day information from the Grand Master node.

The fundamental concept is to be able to time stamp the PTP frames with high precision as close to the physical wires as possible.

In order to support PTP protocol, the device decodes the EtherType/Sub-type fields from a packet and recognizes that these are special PTP messages which need to be forwarded to the CPU. The device also time stamps these 802.1AS control frames when they arrived into a given node and when they depart from a node. The device supports full flexibility to configure any of the 16 PTP frame types to be time stamped using MsgIdTSEN (PTP Global register offset 0x1). The PTP frame type is determined by the MsgId filed in the PTP Common Header as specified in [Figure 39](#).

The PTP core snoops the frames and based on EtherType and Sub-type fields from the header, determines whether the frame's time stamp information needs to be validated or not. Note that only frames with PTP event messages get time stamped in the PTP core. The device supports two arrival counters and one departure counter. This ensures that more than one arriving event message's time stamp can be captured in hardware. For example, a sync frame coming from a Grand Master may arrive around the same time as when PDelayReq or PDelayResponse message arrives into a given node as there is no time correlation between these two types of PTP frames.

The switch data pipe recognizes the reserved multicast destination address used by the PTP frame and forwards it to the CPU_DEST (Global offset 0x1A). The received PTP frame does not get modified before being sent to the CPU_DEST, except for adding a To_CPU DSA tag. The device supports a two-step PTP clock, wherein a follow-up message is sent out by the software to communicate the residence time¹ of this node. When PTPArrIntEn (PTP Global Config register 0x03) is set to 0x1, an interrupt gets generated by the device whenever a PTP event message has been time stamped by the hardware. PTPArr0IntStatus (PTP Port Status Register offset 0x0) and/or PTPArr1IntStatus (PTP Port Status register offset 0x4) specifies whether any error condition has been triggered during the process of collecting this arrival time stamp. Upon successfully capturing the time stamp in the time stamp register PTPArr0Time (PTP Port Status Register Offset 0x01 & 0x02) or PTPArr1Time (PTP Port Status Register Offset 0x05 & 0x06), the corresponding valid bit PTPArr0TimeValid (PTP Port Status Register Offset 0x0) or PTPArr1TimeValid (PTP Port Status Register Offset 0x04) bit gets set. Similar to the PTP arrival interrupt and time stamp registers, there are PTP frame departure interrupt and time stamp registers supported by the device. The device also captures the sequence identifier from the PTP Common header ([Figure 39](#)) for both arrival and departure PTP frames, which ensures that software always deciphers the time stamp information for the correct PTP frame.

Once the PTP software receives the frame, several fields including the residence time may need to be updated and the frame may need to be forwarded towards a downstream PTP slave node. The PTP frames arriving into hardware from the software are labeled as From_CPU DSA tagged frames.

1. The residence time is the amount of time elapsed from the point the PTP frame entered this node on the physical wires to when it actually got sent out to the downstream node.



Link Street®

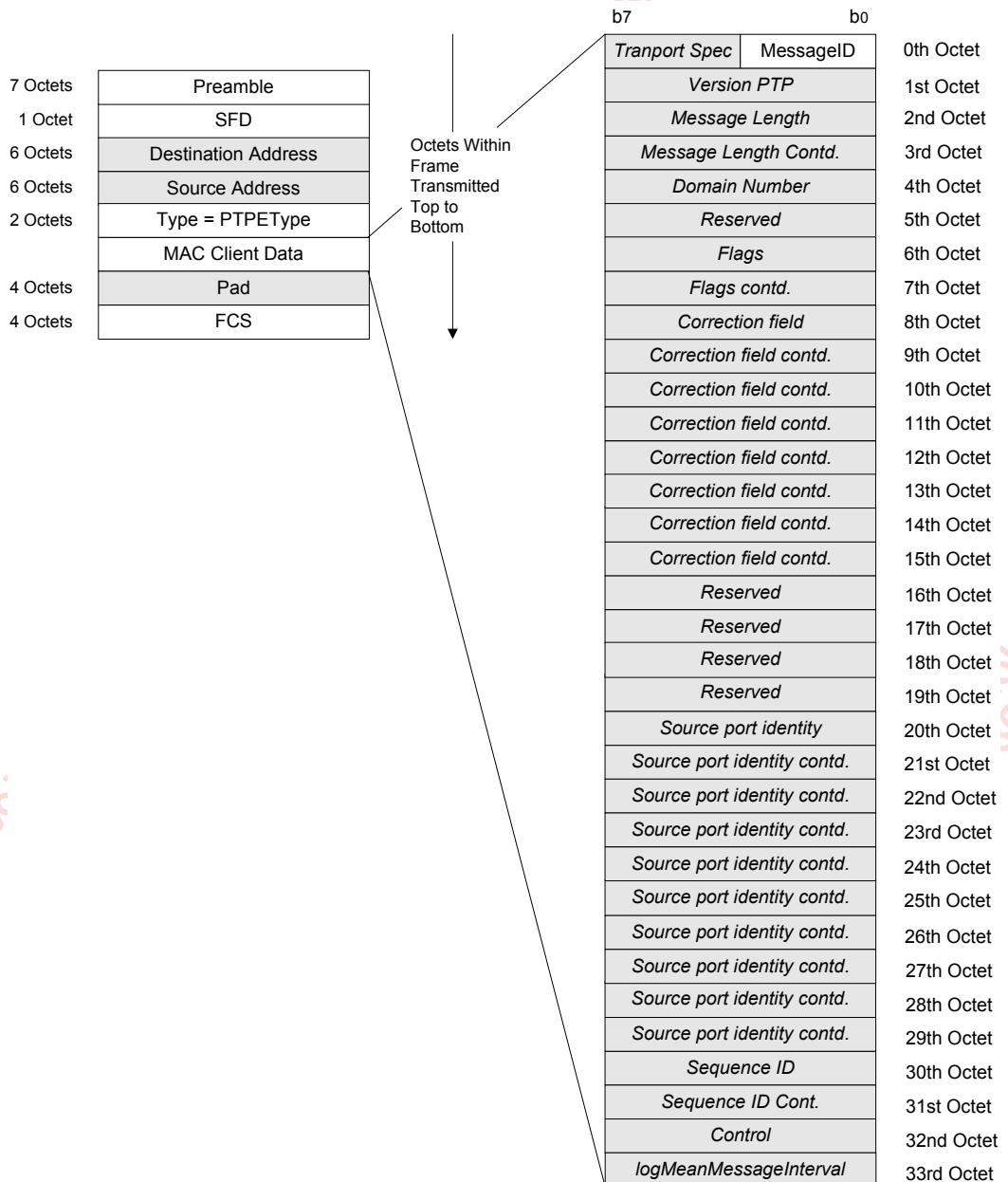
88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Note that a per port PTPInt (PTP Global Status Register 0x08) bit gets set whenever an incoming PTP frame is time stamped and PTPArrIntEn is set for that port or when an outgoing PTP frame is time stamped and PTPDepIntEn is set for that port. The interrupt bit gets cleared after the software reads and clears PTPArr0TimeValid (PTP Port Status register offset 0x0) or PTPArr1TimeValid (PTP Port Status register offset 0x04) for ingress PTP frame that requires time stamping related interrupt and after software clears PTPDepTimeValid (PTP Port Status register offset 0x08) for the egress PTP frame that requires time stamping.

As the PTP frame flows through the switch data path on ingress it may get discarded due to queue congestion, policy, CRC, etc., reasons and on egress may get discarded due to CRC or policy reasons. The hardware keeps track of such discard events by incrementing PTPTSArrDisCtr (PTP Port Status register offset 0x0D) or PTPNonTSArrDisCtr (PTP Port Status register offset 0x0D). Where PTPTSArrDisCtr is incremented for PTP frames that require time stamping and PTPNonTSArrDisCtr is incremented for PTP frames that do not require time stamping. Similarly, there are two departure counters supported in hardware namely PTPTSDepDisCtr (PTP Port Status register offset 0x0D) and PTPNonTSDepDisCtr (PTP Port Status register offset 0x0D).

Figure 39: PTP Common Header Format





Link Street®

88E6321/88E6320 Functional Specification

6.12 Interrupt Controller

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

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

- The EEInt indicates the processing of the EEPROM contents is complete and the I/O registers are now available for CPU access. A CPU can use this interrupt to know it is OK to start accessing the device's registers. The EEInt will assert the device's INT pin even if not EEPROM is attached unless the EEPROM changes the contents of the EEIntMast register (Global 1, offset 0x04).
- The StatsDone, VTUDone and ATUDone interrupts de-assert after the Switch Global Status register is read (these interrupt status bits are clear on read). These interrupts indicate that the last Stats operation, VTU operation and/or ATU operation has completed.
- The PHYInt indicates that one or more of the PHY interrupts enabled in PHY register offset 0x12 are active. The PHYInt will stay low until the PHY interrupts are serviced (see the PHY functional description on how to process its interrupts – See PHYInt bit [Table 121](#)).
- The DeviceInt indicates that the source of the interrupt is from the Interrupt Source register (Global 2 offset 0x00). The DeviceInt will stay low until the source of the interrupt is serviced.

6.12.1 Device Interrupts

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

- A WatchDog event occurred. WatchDog events are enabled in Global 2 offset 0x1B.
- A Jam Limit event occurred ([Section 2.3.6](#)).

7

Accessing Data Structures

The device contains many data structures that are used to control switching. The larger structures have specialized ways to access them and they are capable of generating an interrupt to the CPU if they need servicing. These larger structures are the Ternary Content Addressable Memory (TCAM - [Section 7.1](#)), the VLAN Databases controlled by the VTU ([Section 7.2](#)), and the Address Database controlled by the ATU ([Section 7.3](#)).

7.1 Ternary Content Addressable Memory (TCAM) - (88E6321 Only)

The Ternary Content Addressable Memory (TCAM) in the device supports user commands to access and modify the contents of the TCAM entries. This section focuses on the TCAM's structure and how to access its contents.

All TCAM operations have the same user interface and protocol. Global 3 registers are used and are shown in [Table 21](#). The protocol for a TCAM operation is as follows:

- Ensure the TCAM is available by checking the TCAMBusy bit in the TCAM Operation register. The TCAM can only perform one user command at a time.
- Load the TCAM Keys, Frame Data or Action register values that are to be associated with this TCAM entry and selected TCAM Page if a TCAM Load operation is being done. When loading a TCAM entry, there is a recommended order that the TCAM's Keys, Data, and Actions should be entered.
- Start the TCAM operation by defining the required TCAMOp, TCAM Page, TCAM Entry and setting the TCAMBusy bit to a one in the TCAM Operation register – this can be done with a single write operation.
- Wait for the TCAM operation to complete. This can be done by polling the TCAMBusy bit in the TCAM Operation register.
- Read the results if appropriate.

Table 21: TCAM Operation Register

Register	Global 3 Offset	Before the Operation Starts	After the Operation Completes
TCAM Operation	0x00 Page 0, 1, and 2	Used to define the required operation (including which Entry and Page to act on) and start it.	Used to indicate the TCAM's Busy status.
TCAM Keys (4 registers)	0x02 to 0x05 Page 0	Used to validate the entry (Valid) and define the entry's parameters.	Used to return the Key data found in the selected entry.
TCAM Frame Data (48 registers)	0x06 to 0x1B Page 0 0x02 to 0x1B Page 1	Used to define up to 48 bytes of Frame Data used to match to the ingressing frames (two TCAM entries can match up to 96 bytes).	Used to return the Frame data found in the selected entry.
TCAM Actions (4 registers)	0x02 to 0x05 Page 2	Used to define the Action that is applied to any frame that matches the Key & Frame Data as masked.	Used to return the Action data found in the selected entry.

7.1.1

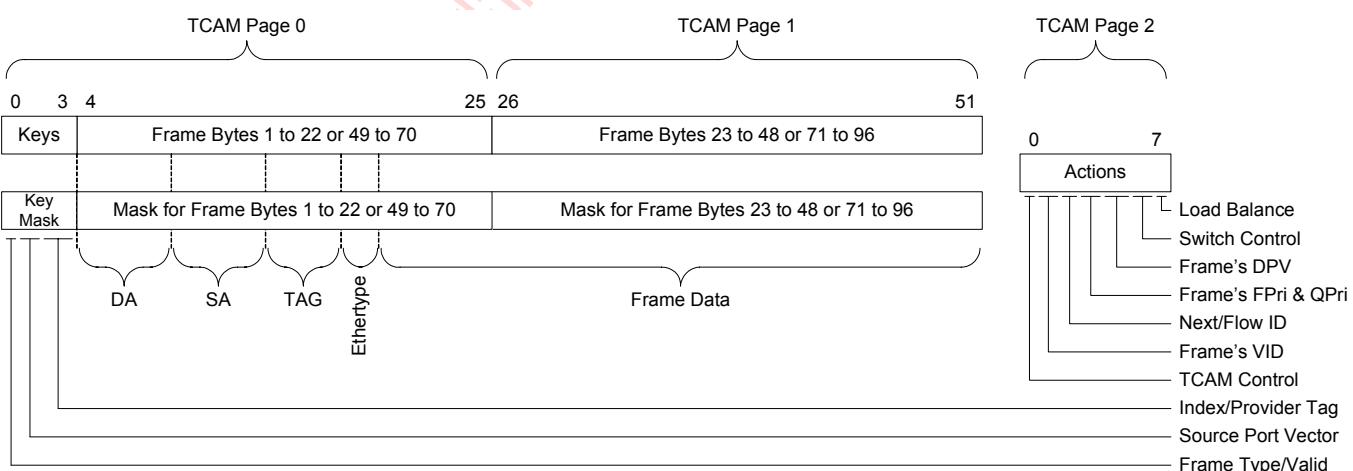
Format of the TCAM Database

Each TCAM entry in the database contains:

- A 4 byte Key and Mask used to validate the entry (Valid) and define the entry's parameters,
- 48 bytes of Frame Data and Mask used to bit-wise match to ingressing frames, Two TCAM entries can be linked together to match up to 96 bytes deep into a frame.
- 4 bytes of Action data that is applied to any frame that matches the Key & Frame Data as masked

The format of a TCAM entry is shown in [Figure 40](#) and [Table 22](#) and [Table 23](#). The database is accessed 16-bits at a time via the Switch Global registers shown in [Table 21](#) (not all the register bits are shown). For more information about these register see the TCAM Operation register (Global 3 offset 0x00) and TCAM Data registers (Global 3, Pages 0, 1 and 2, offsets 0x02 to 0x1B).

Figure 40: Format of a TCAM Entry



The TCAM Key and Frame Bytes (those shown in TCAM Page 0 and TCAM Page 1 in the above figure) are used to bit-wise match to ingressing frame data. A three state matching approach is used. Any bit can be matched to zero, matched to one, or not matched at all (i.e., the value in the frame's bit is don't care). In order to support three states, two bits are needed. The top line in the figure contains the data bits to match and the bottom line contains the Mask bits that are associated with each data bit.

The 16-bit registers that are used to access the TCAM are organized such that the data bits reside in the lower 8 bits of each 16-bit register. The upper 8 bits of each register are the Mask bits for the lower 8 bits where bit 15 is the mask for bit 7, bit 14 is the mask for bit 6, etc. The individual pairs of data bits and mask bits work together as follows:

Mask	Data	Meaning
0	0	Don't Care. The data bit can be a one or a zero for a TCAM hit to occur.
1	0	Hit on 0. The frame's data bit must be a zero for a TCAM hit to occur.
1	1	Hit on 1. The frame's data bit must be a one for a TCAM hit to occur.
0	1	Never Hit. Used to prevent a TCAM hit from occurring from this entry. This is needed so that a TCAM entry can be defined as unused or invalid.

The Never Hit value is used to Flush the TCAM or Purge a TCAM entry and should be limited to being used on Key byte 0 only (TCAM Page 0, offset 0x02).

Table 22: TCAM Entry Format - TCAM Key and Frame Data with Masks

Field	Bits	Description
Frame Type/Valid	7:6 in offset 0x02, Page 0 15:14 = Mask	Frame Type and Entry Valid. These bits are used to indicate which frame types can match this TCAM entry. Supported frame types are Normal, Normal supports both Tagged and Untagged frames. DSA Tagged or Provider Tagged. These bits are also used to invalidate unused TCAM entries so they are not used for matching.
Source Port Vector	6:0 in offset 0x03, Page 0 14:8 = Mask	Source Port Vector. These bits are used to associate a TCAM entry with one, or more than one, ingress ports. It essentially indicates which ingress ports are not allowed to use this TCAM entry.
Index/Provider Tag	Offset 0x04 and 0x05, Page 0 15:8 = Mask	Index or Provider Tag. If this is a TCAM entry for the 2nd 48 bytes of a frame (for 96 byte deep matches), offset 0x05 contains the non-zero Index value that must match the Next value returned from the 1st 48 byte TCAM hit (see Next/Flow ID bits below and Section 7.1.3.2 on how to create 96-byte deep TCAM matches). If this is a TCAM entry for the 1st 48 bytes of a frame and if the Frame Type above is Provider, these bits contain the Provider's frame Priority and VID bits to match. Otherwise these bits must be zero.
Frame Bytes 1 to 22 or 49 to 70	Offset 0x06 to 0x1B, Page 0 15:8 = Mask	Frame Bytes to match. If this is a TCAM entry for the 1st 48 bytes of a frame, these bytes are used to match bytes 1 to 22 of the frame. If this is a TCAM entry for the 2nd 48 bytes of a frame (for 96 byte deep matches), these bytes are used to match bytes 49 to 70 of a frame. NOTE: All frames are considered 802.1Q Tagged during this compare even if the ingressing frame isn't physically tagged. This approach allows one TCAM entry to match both tagged or untagged frames.
Frame Bytes 23 to 48 or 71 to 96	Offset 0x02 to 0x1B, Page 1 15:8 = Mask	Frame Bytes to match. If this is a TCAM entry for the 1st 48 bytes of a frame, these bytes are used to match bytes 23 to 48 of the frame. If this is a TCAM entry for the 2nd 48 bytes of a frame (for 96 byte deep matches), these bytes are used to match bytes 71 to 96 of a frame.



Link Street®

88E6321/88E6320 Functional Specification

Table 23: TCAM Entry Format - Action Data

Field	Bits	Description
TCAM Control	15:13 in offset 0x02, Page 2	TCAM Control. These Action bits are used to indicate that a match on the 1st 48 bytes of a frame needs to Continue on to match the 2nd 48 bytes of the frame before a TCAM Hit can be considered. Other bits can be used to increment the TCAM Counter (Port offset 0x12) if enabled, or cause a TCAM interrupt (Global 1 offset 0x00).
Frame's VID	12:0 in offset 0x02, Page 2	Frame's VID. On a TCAM Hit, these Action bits are used as the VID assigned to the frame if bit 12 of this field is a one. This VID will be used for VTU and then ATU lookups and the frame will be transmitted with this VID if it egresses a port Tagged (IEEE 802.1Q tagged, Provider Tagged or DSA Tagged).
Next/Flow ID	15:8 in offset 0x03, Page 2	Next or Flow ID. On a match of the 1st 48 bytes of a frame where the Continue bit = 1 (one of the TCAM Control bits above), these bits contain the non-zero Next Index value than needs to match the TCAM's Key Index bits (above) in order to get a match on the 2nd 48 bytes of a frame (see Section 7.1.3.2 on how to create 96-byte deep TCAM matches). On a TCAM Hit, these Action bits are used as the Flow ID assigned to the frame. The lower two bits for the Flow ID are sent to the Port Ingress Rate Limiter (PIRL – Section 3.5) block so that per flow limiting can be done.
Frame's FPri and QPri	7:0 in offset 0x03, Page 2	Frame's assigned Frame Priority and Queue Priority. On a TCAM Hit, Action bits 2:0 are used as the FPri assigned to the frame if bit 3 of this field is a one. The frame will be transmitted with this FPri if it egresses a port Tagged (IEEE 802.1Q tagged, Provider Tagged or DSA Tagged). On a TCAM Hit, Action bits 5:4 are used as the QPri assigned to the frame if bit 7 of this field is a one. The frame will be placed into the egress queue defined by this QPri.
Frame's DPV	11, 6:0 in offset 0x04, Page 2	Frame's Destination Port Vector. On a TCAM Hit, Action bits 6:0 are used as the DPV assigned to the frame if bit 11 of this field is a one. The frame will be transmitted out the ports where the DPV is a one (Port 0 is controlled by bit 0, Port 1 by bit 1, etc.).
Switch Control	15:4 in offset 0x05, Page 2	Switch Control bits. On a TCAM Hit, Action bits 14:4 are used as various Switch Control bits if bit 15 of this field is a one. The Switch Control bits can be used to add a Tag to a frame that ingresses tagged and to force egress to use the VID assigned to the frame instead of the ingress port's DefaultVID (needed for Provider egress). They can also define a frame to be Management (MGMT), an Arp (ARP), a Snoop, a PolMirror or a PolTrap. And SA and/or DA Non Rate Limiting can be enabled as well (DaNRL and SaNRL are used by the Port Ingress Rate Limiter, PIRL, Section 3.5). See each bits description in Global 3, Page 2 offset 0x05.
Load Balance	3:0 in offset 0x05, Page 2	Trunking Load Balance value. On a TCAM Hit, Action bits 2:0 are used as the Load Balance value assigned to the frame if bit 3 of this field is a one. The Load Balance value is used to access the Trunk Mask Table (Global 2 offset 0x07) as shown in Section 10.4.2 .

7.1.2 Reading the TCAM Database

7.1.2.1 Dumping or Searching for Valid TCAM Entries

The contents of the TCAM database can be dumped or searched. The dump operation is called TCAM Get Next since it returns the active contents of the TCAM database in ascending TCAM Entry order. Invalid or unused entries are skipped. A search operation can also be done using the TCAM Get Next operation.

The TCAM Get Next operation starts with the entry contained in the TCAM Entry register and returns the next higher active TCAM Entry in the TCAM database. Page 0 must be used for all TCAM Get Next operations. Use a TCAM Entry of all ones to get the first or lowest active TCAM Entry. The returned TCAM Entry's number and its Page 0 data are accessible in the TCAM Data registers. To access to the other Pages of the TCAM Entry use the TCAM Read operation (Section 9.1.2.2). To get the next higher active entry, start the TCAM Get Next operation again with the last TCAM Entry register's value since it was the last valid entry. A returned TCAM Entry of all ones indicates that no higher active TCAM Entry was found or that the TCAM Entry value of 0xFF was found. In either case, it indicates that the end of the database has been reached. If it were reached with a valid TCAM Entry of 0xFF the entry's Page 0, offset 0x02 will not equal 0x00FF (a value of 0x00FF in Page 0 offset 0x02 indicates an invalid or unused entry). A summary of how the TCAM Get Next operation uses the TCAM's registers is shown in [Table 24](#).

Table 24: TCAM Get Next Operation Register Usage

Register	Global 3 Offset	Before the Operation Starts	After the Operation Completes
TCAM Operation	0x00 Page 0	Used to define the required operation (including which Entry and Page to act on) and start it. TCAM Get Next operations must be done with TCAM Page equal to 0x0.	Used to indicate the TCAM's Busy status and return the next higher active Entry if found, or all ones are returned indicating the end of the table has been reached (all ones is a valid entry if Page 0 offset 0x02 is not equal to 0x00FF).
TCAM Keys (4 registers)	0x02 to 0x05 Page 0	Ignored	If offset 0x02 equals 0x00FF the entry is not valid. Otherwise, the TCAM Keys can be read.
TCAM Data (22 registers)	0x06 to 0x1B Page 0	Ignored	If offset 0x02 equals 0x00FF the entry is not valid. Otherwise, the 22 bytes of the TCAM Data can be read. See Section 9.1.2.2 on how to read the other Pages of data for this TCAM entry.

To search for a particular TCAM Entry, start the TCAM Get Next operation with a TCAM Entry one less than the target TCAM Entry. If the target TCAM Entry is found, it is returned in the TCAM Entry register along with its associated Page 0 data in the TCAM Data registers. If the target TCAM Entry is not found active, the TCAM Entry register contents will not equal the target TCAM Entry.

7.1.2.2 Reading Arbitrary TCAM Entries or Pages

After reading the contents of one Page of data from a TCAM entry, the Read TCAM operation can be used to load the Global 3 registers with the another Page of data from the same TCAM Entry or another one. A summary of how the TCAM Read operation uses the TCAM's registers is shown in [Table 25](#).



Link Street®

88E6321/88E6320 Functional Specification

Table 25: TCAM Read Operation Register Usage

Register	Global 3 Offset	Before the Operation Starts	After the Operation Completes
TCAM Operation	0x00 Page 0, 1, or 2	Used to define the required operation (including which Entry and Page to act on) and start it. TCAM Read operations can use any valid Page number and that Page's data will be returned.	Used to indicate the TCAM's Busy status.
TCAM Keys (4 registers)	0x02 to 0x05 if Page 0	Ignored	Returns the TCAM Keys for the requested entry if Page equals 0.
TCAM Data (48 registers)	0x06 to 0x1B if Page 0, else 0x02 to 0x1B if Page 1	Ignored	Returns the TCAM Match Data if the requested Page equals 0 or 1. The range of valid registers depends on the Page number.
TCAM Actions (4 registers)	0x02 to 0x05 if Page 2	Ignored	Returns the TCAM Action data if the request Page equals 2.

A complete dump of all valid TCAM contents can be done by using the TCAM Get Next to find the valid entries and its Page 0 data, then use the TCAM Read to get to that valid entries Page 1 and Page 2 data. Then a TCAM Get Next can be used again to find the next valid entry, etc.

7.1.3

Loading an Entry in the TCAM Database

Any TCAM Entry can be loaded into the VLAN TCAM database by using the TCAM Load operation. An entry is loaded into the database if none of the TCAM Match bits for the entry have the Never Hit combination (see Section 9.1.1).



Note

NOTE: The Never Hit combination of bits is used to invalidate a TCAM entry and should only be used in an entry's 1st Key Data byte (i.e., by setting TCAM Page 0, offset 0x02 to 0x00FF). The TCAM GetNext operation skips the reading of any entry whose Page 2, offset 0x02 equals 0x00FF.

The Load operation accesses the TCAM database using the TCAM Entry and TCAM Page registers. Since each TCAM entry spans across multiple Pages, the order in which the TCAM is loaded is important if the TCAM's contents are being modified while data is flowing through the switch and the TCAM is enabled for use on those active ports. Basically, the entry needs to be loaded from back to front as defined below.

7.1.3.1

Example of Loading a 48 Byte TCAM Entry

A 48 Byte TCAM entry is loaded as follows:

1. Choose any unused 48 byte TCAM entry or any entry that can be replaced.
 2. Flush the chosen entry using the TCAM Flush a single TCAM entry command (see section 9.1.5).
 3. Using the selected entry, load the Action data (TCAM Page 2) for the entry. This Action data is the desired overrides or actions that are applied to frames that match this 48 byte TCAM entry (i.e., the entry's Continue bit must be zero). Only those action bits that need to set to a one need to be loaded assuming the Flush in above step was done.
 4. Continuing with the selected entry, load the Frame Data and Masks for frame bytes 23 to 48 (TCAM Page 1). Only those bytes with bits that need to match to specific values in the frames need to be written to as the Flushed value for these bytes is all “Don’t Care”.
 5. Continuing with the selected entry, load the Key Data & Frame Data (and Masks) for frame bytes 1 to 22. The Key Data is used to define the Frame Type this entry will match, the source ports this entry must not be used on, and Provider Tag information if the entry is for Provider Tagged frames. All unused Key Data bytes and Frame Data bytes can be left alone in their Flushed state if they are not needed. This completes the loading of the entire TCAM entry and will be applied immediately to all applicable frames.

7.1.3.2

Loading a 96 Byte TCAM Entry

A 96 Byte TCAM entry is loaded as follows:

1. Choose two unused 48 byte TCAM entries or two 48 byte entries or one 96 byte entry that can be replaced.
 2. Flush the chosen entries using the TCAM Flush a single TCAM entry command (see section 9.1.5).
 3. Using the higher numbered entry, load the Action data (TCAM Page 2) for the entry. This Action data is the desired overrides or actions that are applied to frames that match this 96 byte TCAM entry (i.e., the entry's Continue bit must be zero). Only those action bits that need to be set to a one need to be loaded assuming the Flush in above step was done.
 4. Continuing with the higher numbered entry, load the Frame Data and Masks for frame bytes 71 to 96 (TCAM Page 1). Only those bytes with bits that need to match to specific values in the frames need to be written to as the Flushed value for these bytes is all "Don't Care".



Link Street®

88E6321/88E6320 Functional Specification

5. Continuing with the higher numbered entry, load the connection Key Data & Frame Data (and Masks) for frame bytes 49 to 70. The connection Key Data is used to connect this entry of frame bytes 49 to 96 with the TCAM entry for frame bytes 1 to 48 (which will be loaded later). This connection is made using this entry's Index field (TCAM Page 0 offset 0x05) where a good default value to use is this entry's TCAM Entry number – see [Section 7.1.1](#) for other possibilities. The entry's Frame Type must also be set to 0x0 and all other Key Data bytes and Frame Data can be left alone in their Flushed state if they are not needed. This completes the loading of the TCAM entry for the 2nd 48 bytes of the frame.
6. Using the lower numbered entry, load the Action data (TCAM Page 2) for this entry. This Action data must be the connection mechanism between the 1st and 2nd 48-byte TCAM entries that form a complete 96-byte TCAM entry. The connection is made by setting this entry's Next Index field (TCAM Page 2 offset 0x03) to the same value of the Index field loaded in the step above. This entry's Continue bit must also be set to a one and the rest of the entry's Action bits should all be left zero (their values after a Flush).
7. Continuing with the lower numbered entry, load the Frame Data and Masks for frame bytes 23 to 48 (TCAM Page 1). Only those bytes with bits that need to match to specific values in the frames need to be written to as the Flushed value for these bytes is all "Don't Care".
8. Continuing with the lower numbered entry, load the Key Data & Frame Data (and Masks) for frame bytes 1 to 22. The Key Data is used to define the Frame Type this entry will match, the source ports this entry must not be used on, and Provider Tag information if the entry is for Provider Tagged frames. All unused Key Data bytes and Frame Data bytes can be left alone in their Flushed state if they are not needed. This completes the loading of the entire TCAM entry and will be applied immediately to all applicable frames.

A summary of how the Load operation uses the TCAM's registers is shown in [Table 26](#).

Table 26: TCAM Load Operation Register Usage

Register	Global 3 Offset	Before the Operation Starts	After the Operation Completes
TCAM Operation	0x00 Page 0, 1, or 2	Used to define the required operation (including which Entry and Page to act on) and start it. TCAM Read operations can use any valid Page number and that Page's data will be returned.	Used to indicate the TCAM's Busy status.
TCAM Keys (4 registers)	0x02 to 0x05 if Page 0	Used to define the entry as Valid and define the entry's parameters including Frame Type and applicable source ports.	No change.
TCAM Data (48 registers)	0x06 to 0x1B if Page 0, else 0x02 to 0x1B if Page 1	Used to define each bit the 1st 48 or 96 bytes of Frame Data that is to match the ingressing frames (two TCAM entries can match up to 96 bytes). Each bit can match to a one, to a zero or don't care.	No change.
TCAM Actions (4 registers)	0x02 to 0x05 if Page 2	Used to define the Actions that are applied to any frame that matches the Key & Frame Data as masked.	No change.



Note

A load operation will not load a 'valid' usable entry if any of its TCAM Match bits have a Never Hit combination (Section 9.1.1).

7.1.4

Moving an Entry

If more than one TCAM Entry results in a match or 'hit' for a frame, the lowest numbered TCAM Entry's Actions will be selected. So if longest prefix matching is being done, successively longer match entries must appear in the TCAM with a lower Entry number compared to the successively shorter match entries.

Due to the above, existing TCAM entries may need to be moved in the table to make room for the proper insertion of new entries. And this needs to occur will frames are flowing such that all existing TCAM entries result in valid 'hits'. This means the move needs to be done such that the existing entries always exist in the TCAM. This can be done as follows:

7.1.4.1

Example of Moving a TCAM Entry While it is Being Used

A TCAM entry is moved as follows:

1. Choose the new location or locations for the TCAM entry that needs to be moved.
2. Load a copy of the existing TCAM entry in the new location(s) using the methods defined in 4.1.3.
3. At this time two identical copies of this TCAM entry exist in the TCAM. Both are 'hit' when a frame matches, but the lowest entry's Actions are used. Since the Actions are identical on both entries the desired Actions are performed on the frame.
4. Flush the original entry using the TCAM Flush a single TCAM entry command (see section 9.1.5). This removes the original, making the copy the new active entry. This completes the Moving of a TCAM entry.

Any TCAM Entry can be moved higher up or lower down in the TCAM using the above method. Since a 'make before break' approach is used, frames continue to match and get the intended TCAM Actions applied to them during this process.

7.1.5

Updating an Entry

A TCAM Entry's Actions can be updated by loading a new set of Actions into that entry as follows:

1. Read Page 2 (the Actions) on the TCAM entry that needs to be updated. Use the Read TCAM Operation.
2. Update the just read data as needed where appears in the Global 3 register space. Only those action bits that need to updated need to be modified assuming the Read in above step was done.
3. Load the new set of Action data (TCAM Page 2) for the entry. Use the Load TCAM Operation on Page 2. This Action data is the updated desired overrides or actions that are applied to frames that match this TCAM entry (i.e., the entry's Continue bit must be zero).

7.1.6

Flushing Entries

All Entries in the TCAM database can be purged by a singe Flush All Entries TCAM Operation. None of the other TCAM registers are used by the Flush All Entries command.



Link Street®

88E6321/88E6320 Functional Specification

7.2

VLAN Translation Unit Operations

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

All VTU operations have the same user interface and protocol. Global 1 registers are used and are shown in [Table 27](#). The protocol for an VTU operation is as follows:

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

Table 27: VTU Operation Register

Register	Global 1 Offset	Before the Operation Starts	After the Operation Completes
VTU FID	0x02	Used to define which address database is to be associated with this VID and if this VID is a Policy VID.	Use to indicate the returned VID's FID and Policy.
VTU SID	0x03	Used to define which 802.1s instance is to be associated with this VID	Used to indicate the returned VID's SID
VTU Operation	0x05	Used to define the required operation (including which database to associate with this VID) and start it.	Used to indicate the VTU's Busy status and violation status including source of the violation.
VTU VID	0x06	Used to further define the required operation and used as the VID that is to be operated on.	Returns the VID from the desired operation.
VTU Data (3 registers)	0x07 to 0x09	Used to define the required data that is to be associated with the required VID, including VTU Priority Override.	Returns the associated data from the desired operation.

7.2.1 Format of the VTU Database

Each VID entry in the VTU database contains:

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

The format of a VTU entry is shown in [Figure 41](#) and [Table 28](#). The database is accessed 16-bits at a time via the Switch Global registers shown in [Table 28](#) (not all the register bits are shown). For more information about these register see the VTU Operation register (Global 1 offset 0x05) and VTU Data registers for all ports (Global 1 offsets 0x02, 0x03 and 0x06 to 0x09).

Figure 41: Format of a VTU Entry

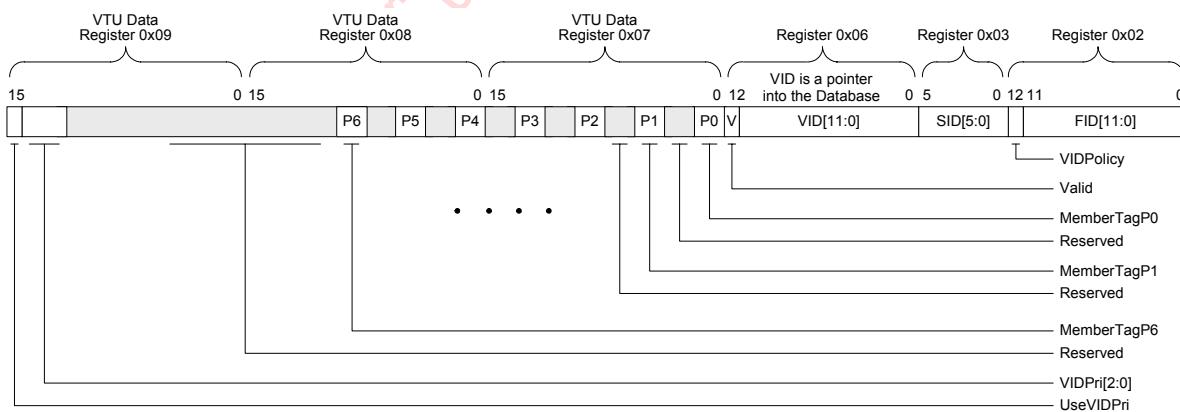


Table 28: VTU Entry Format

Field	Bits	Description
UseVIDPri	15 in Reg 0x09	VID Priority Override. This bit is used to indicate that frames assigned with this VID can have their priority overridden with the VIDPri bits below if either of the port's VTUPriOverride bits is set (see Port offset 0x08).
VIDPri	14:12 in Reg 0x09	VID Priority override value when enabled by the UseVIDPri bit above. Used for priority override on ingressing frames (see Section 3.4.6). Enabling a priority on a VID will override the frame's priority only if the port's VTUPriOverride bits are configured to do so (Port offset 0x0C).
Reserved for 802.1s Port State	In Reg 0x07: Port 0 3:2 Port 1 7:6 Port 2 11:10 Port 3 15:14 In Reg 0x08: Port 4 3:2 Port 5 7:6 Port 6 11:10	These bits are NOT part of the VTU and are NOT associated with the VID. They are used to access the STU database (802.1s per VLAN spanning tree – Section 3.2.3). On writes to the VTU these bits are don't care. On reads from the VTU these bits will not be updated and will contain the data from the last STU read (Section 7.2.7).



Link Street®

88E6321/88E6320 Functional Specification

Table 28: VTU Entry Format

Field	Bits	Description
MemberTag	In Reg 0x07: Port 0 1:0 Port 1 5:4 Port 2 9:8 Port 3 13:12 In Reg 0x08: Port 4 1:0 Port 5 5:4 Port 6 9:8	The lower two bits of each port's VTU data is called MemberTag. These bits are used to indicate which ports are members of the VLAN and if these VLANs frames should be tagged or untagged, or unmodified when exiting the port as follows 00 = Port is a member of this VLAN and frames are to egress unmodified 01 = Port is a member of this VLAN and frames are to egress Untagged 10 = Port is a member of this VLAN and frames are to egress Tagged 11 = Port is not a member of this VLAN
Valid	12 in Reg 0x6	Valid bit. This bit is used to indicate that the below VID and its associated data is valid in the VTU's database and should be used. After a hardware reset, all 4096 entries in the table are considered invalid (the Valid bit on each entry is cleared).
VID	11:0 in Reg 0x6	VLAN ID. These bits indicate the VID number that is associated with the MemberTag data, VTU Priority and its override, VTU Policy and the entry's Forwarding Information Database number (FID).
SID	5:0 in Reg 0x03	802.1s Information Database Number. If 802.1s per VLAN spanning tree is being used, these bits indicate the spanning tree instance number to use for all frames assigned with this VID (see Section 3.2.3). Multiple VID's can use the same SID. If per VLAN spanning trees are not used the SID must be written as zeros.
VIDPolicy	12 in Reg 0x02	VID Policy Entry. This bit is used to indicate that frames assigned with this VID can have Layer 2 Policy actions applied to it if either of the port's VTU Policy bits is set (see Section 3.1.3.1 and Port offset 0xE).
FID	11:0 in Reg 0x02	Forwarding Information DataBase Number. If separate address databases are used, these bits indicate the address database number to use for all frames assigned with this VID (see Section 2.4.8). All MAC DA look-ups and SA learning will refer to the address database number defined by the FID associated with the frame's VID. Multiple VID's can use the same FID. If separate address databases are not used the FID must be written as zeros.

7.2.2

Reading the VLAN Database

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

The VTU Get Next operation starts with the VID contained in the VTU VID register and returns the next higher active VID in the VLAN database. Use a VID of all ones to get the first or lowest active VID. The returned VID and its data are accessible in the VTU Operation, VTU VID, VTU FID, VTU SID and the VTU Data registers. To get the next higher active VID, the VTU Get Next operation can be started again without setting the VID registers since it already contains the last address. A returned VID of all ones indicates that no higher active VID was found or that the VID value of 0xFFFF was found. In either case, it indicates that the end of the database has been reached. If it were reached with a valid VID of 0xFFFF the entry's Valid bit is returned set to one. A summary of how the VTU Get Next operation uses the VTU's registers is shown in [Table 29](#).

Table 29: VTU Get Next Operation Register Usage

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

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



Link Street®

88E6321/88E6320 Functional Specification

7.2.3

Loading and Purguing an Entry in the VLAN Database

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

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

**Note**

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

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

A summary of how the Load operation uses the VTU's registers is shown in [Table 30](#).

Table 30: VTU Load/Purge Operation Register Usage

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

7.2.4

Flushing Entries

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

**Note**

When the VTU is flushed the STU is also flushed ([Section 7.2.4](#))

7.2.5**Servicing VTU Violations**

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

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

A summary of how the Get/Clear Violation Data operation uses the VTU's registers is shown in [Table 31](#).

Table 31: VTU Get/Clear Violation Register Usage

Register	Global 1 Offset	Before the Operation Starts	After the Operation Completes
VTU Operation	0x05	Used to define the desired operation and start it.	Used to indicate the VTU's Busy status, the type of violation and the source port of the violation.
VTU VID	0x06	Ignored	Used to indicate the VID that was involved in the violation
VTU Data (3 registers)	0x07 to 0x09	Ignored	No change
VTU FID	0x02	Ignored	No change.
VTU SID	0x03	Ignored	No change.

7.2.6

Format of the STU Database

The per VLAN Spanning Tree Unit (STU) in the device supports user commands to access and modify the contents of the Port State database. The description on what an STU is and how to use the STU for switch operations is described in Section 5.2.3. This section focuses on the STU's structure and how to access its contents.

Each STU database entry contains:

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

Each STU entry is accessed by its unique SID. The format of an STU entry is shown in [Figure 42](#) and [Table 32](#). The database is accessed 16-bits at a time via the Switch Global 1 registers shown in [Figure 42](#) (not all the register bits are shown). For more information about these register see the VTU Operation register (Global 1 offset 0x05) and VTU Data registers for all ports (Global 1 offsets 0x03 and 0x06 to 0x09).



The STU is accessed using many of the VTU registers.

Note

Figure 42: Format of an STU Entry

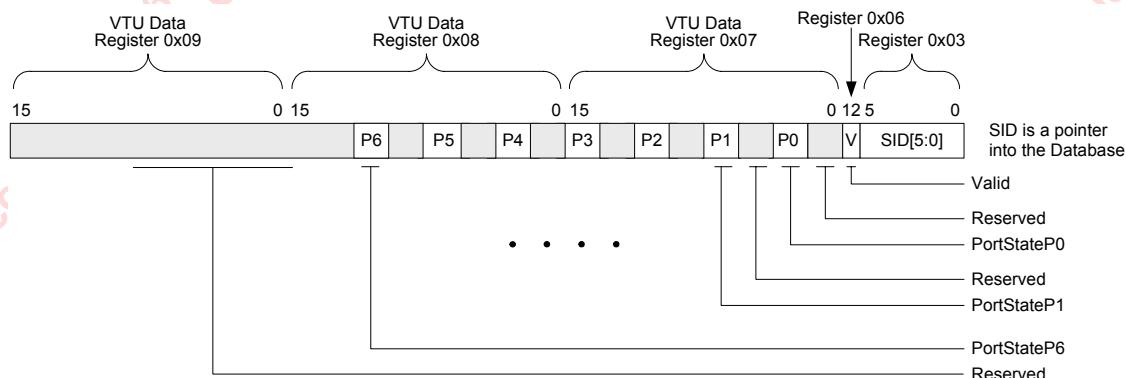


Table 32: STU Entry Format

Field	Bits	Description
Reserved for VTU Priority Override	In Reg 0x09: 15:12	These bits are NOT part of the STU and are NOT associated with the SID. They are used to access the VTU database (802.1Q VLANs – Section 3.2.2). On writes to the STU these bits are don't care. On reads from the STU these bits will not be updated and will contain the data from the last VTU read (Section 7.2.2).
802.1s Port State	In Reg 0x07: Port 0 3:2 Port 1 7:6 Port 2 11:10 Port 3 15:14 In Reg 0x08: Port 4 3:2 Port 5 7:6 Port 6 11:10	The upper two bits of each port's VTU data register is called 802.1s PortState. These bits are used to support 802.1s per VLAN spanning tree as follows: 00 = 802.1s Disabled. Use non-VLAN Port States (i.e., the port's default Port State - Port offset 0x04) for this port for frames with a VID that is associated to this SID 01 = Blocking/Listening Port State for this port for frames with a VID that is associated to this SID 10 = Learning Port State for this port for frames with a VID that is associated to this SID 11 = Forwarding Port State for this port for frames with a VID that is associated to this SID These 802.1s PortState bits take precedence over the port's Port State bits (Port offset 0x04) unless the port's Port State is Disabled (which prevents all frames from flowing).
Reserved for MemberTag	In Reg 0x07: Port 0 1:0 Port 1 5:4 Port 2 9:8 Port 3 13:12 In Reg 0x08: Port 4 1:0 Port 5 5:4 Port 6 9:8	These bits are NOT part of the STU and are NOT associated with the SID. They are used to access the VTU database (802.1Q VLANs – Section 3.2.2). On writes to the STU these bits are don't care. On reads from the STU these bits will not be updated and will contain the data from the last VTU read (Section 7.2.2).
Valid	12 in Reg 0x06	Valid bit. This bit is used to indicate that the below SID and its associated data is valid in the STU's database and should be used. After a hardware reset, all 64 entries in the table are considered invalid (the Valid bit on each entry is cleared).
SID	5:0 in Reg 0x03	802.1S Instance Database number. These bits indicate the SID number that is associated with the 802.1s Port State bits above.



Link Street®

88E6321/88E6320 Functional Specification

7.2.7

Reading the SID Database

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

The STU Get Next operation starts with the SID contained in the VTU SID register and returns the next higher active SID in the STU database. Use a SID of all ones to get the first or lowest active SID. The returned SID and its data are accessible in the VTU Operation, VTU SID and the VTU Data registers. To get the next higher active SID, the STU Get Next operation can be started again without setting the SID registers since it already contains the last address. A returned SID of all ones indicates that no higher active SID was found or that the SID value of 0x3F was found. In either case, it indicates that the end of the database has been reached. If it were reached with a valid SID of 0x3F the entry's Valid bit is returned set to one. A summary of how the STU Get Next operation uses the VTU's registers is shown in [Table 33](#).

**Note**

The STU is accessed using many of the VTU Registers.

Table 33: STU Get Next Operation Register Usage

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

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

7.2.8**Loading and Purging an Entry in the STU Database**

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

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

**Note**

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

**Note**

The STU is accessed using many of the VTU registers.

Table 34: STU Load/Purge Operation Register Usage

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



Link Street®

88E6321/88E6320 Functional Specification

7.2.9

Flushing Entries

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



Note

When the STU is flushed the VTU is also flushed ([Section 7.2.4](#))

7.3**Address Translation Unit Operations**

The Address Translation Unit (ATU) in the device supports user commands to access the contents of the MAC address database. The description on what an STU is and how to use the ATU for switch operations is described in [Table 7.2.6](#). This section focuses on the ATU's structure and how to access its contents.

All ATU operations have the same user interface and protocol. Six Global 1 registers are used and are shown in [Table 35](#). The protocol for an ATU operation is as follows:

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

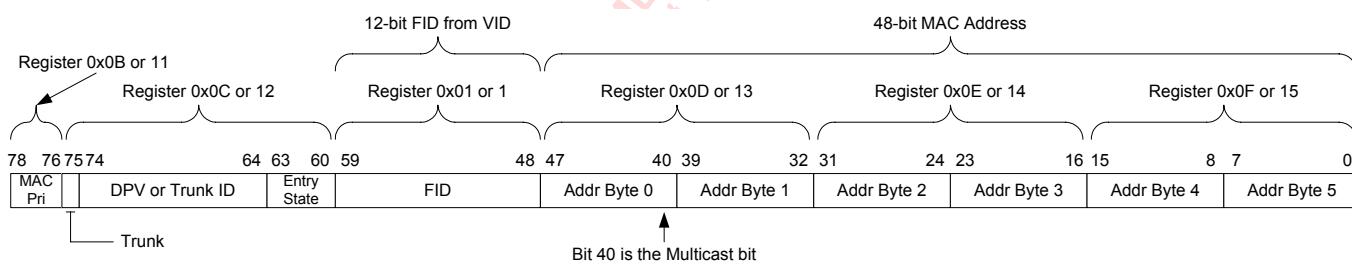
Table 35: Egress Header Fields

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

7.3.1 Format of the ATU Database

Each MAC address entry in the ATU database is 79-bits in size. The lower 48 bits contains the 48-bit MAC address. The next 12 bits contain the FID or Filtering Information Database number, and the upper 19 bits contains information about the entry as shown in [Figure 43](#). The database is accessed 16-bits at a time via the Switch Global 1 registers shown in [Table 36](#).

Figure 43: Format of an ATU Entry



The right 48-bits in [Figure 43](#) is the 48-bit MAC address associated with this ATU entry in the database number defined by the FID. The upper 19 bits is the data that is associated with the entry's MAC address. The upper 19 bits are defined as follows:

Table 36: ATU Data Bits

Field	Bits	Description
MAC Pri	78:76	The MAC's Priority override value when enabled by the EntryState bits below. Used for priority override on ingressing frames (see Section 3.4.6). Enabling a priority on a MGMT MAC address (Multicast EntryState = 0xE or Unicast EntryState = 0xD) will override <i>all</i> priorities for these MGMT frames. Enabling a priority on a static, non-MGMT MAC address, will only override the frame's priority if the port's DAPriOverride and/or SAPriOverride bits are configured to do so (Port offset 0x0C).
Trunk	75	The Trunk bit used to qualify the contents of the DPV or Trunk ID bits below. When this bit is zero bits 74:64 are the DPV (Destination Port Vector) associated with this MAC. When this bit is a one, bits 67:64 is the Trunk ID associated with this MAC (bits 74:68 must be zero in this case).
DPV or Trunk ID	74:64	The Destination Port Vector or Trunk ID. When the Trunk bit (bit 75 above) is a zero these bits indicate which port or ports are associated with this MAC address (i.e., where frames should be switched) when they are set to a one. A DPV of all zeros indicates frames with this DA should be discarded and/or special handling of frames with this SA should occur (see Section 2.4.7 and Section 3.1.2). Bit 64 is assigned to physical Port 0, 65 to Port 1, 66 to Port 2, etc. If more than one port's bit is set to a one frames mapped to this MAC address will attempt to egress out more than one port. This is used for multicast filtering. When the Trunk bit (bit 75 above) is a one bits 67:64 (the lower 4 bits) indicate the Trunk ID that is associated with this MAC address (in this case bits 74:68 must be zeros). The port or ports that this DA MAC address are mapped to is determined by the contents of the Trunk Mapping Table (Global 2 offset 0x08).

Table 36: ATU Data Bits

Field	Bits	Description
EntryState	63:60	<p>The EntryState field, together with the entry's Multicast bit (bit 40) is used to determine the entry's age or its type as follows:</p> <p>For unicast MAC addresses (bit 40 = 0):</p> <ul style="list-style-type: none"> 0x0: Unused entry 0x1 to 0x7: Used entry where EntryState = the Age of the entry where 0x1 is the oldest 0x8: Static Policy entry 0x9: Static Policy entry with Priority Override 0xA: Static Non Rate Limiting (NRL) entry 0xB: Static Non Rate Limiting (NRL) entry with Priority Override 0xC: Static entry defining frames with this DA as MGMT 0xD: Static entry defining frames with this DA as MGMT with Priority Override 0xE: Static entry 0xF: Static entry with Priority Override <p>For multicast MAC addresses (bit 40 = 1):</p> <ul style="list-style-type: none"> 0x0: Unused entry 0x1 to 0x3: Reserved for future use 0x4: Static Policy entry 0x5: Static Non Rate Limiting (NRL) entry 0x6: Static entry defining frames with this DA as MGMT 0x7: Static entry 0x8 to 0xB: Reserved for future use 0xC: Static Policy entry with Priority Override 0xD: Static Non Rate Limiting (NRL) entry with Priority Override 0xE: Static entry defining frames with this DA as MGMT with Priority Override 0xF: Static entry with Priority Override <p>Auto learned entries (Section 2.4.3) will always be unicast entries with an EntryState value in the range of 0x1 (oldest age) to 0x7 (recently added or refreshed).</p> <p>Usage of the various other kinds of EntryState values are covered in the following sections:</p> <ul style="list-style-type: none"> Policy EntryState values is covered in Section 3.1.3 Non Rate Limiting EntryState values is covered in Section 3.5 MGMT (management) EntryState values is covered in Section 6.3.2 Priority Override EntryState values is covered in Section 3.4.5
FID	59:48	Forwarding Information Database number. If multiple address databases are not being used, these bits must remain zero. If multiple address databases are being used, these bits are used to set the desired address database number that is to be associated with this ATU Entry's MAC Address. When frames ingress the switch, the VID assigned to the frame is used to access the VTU. The VTU returns the FID associated with that VID for MAC address lookups in to the ATU.



Link Street®

88E6321/88E6320 Functional Specification

7.3.2

Reading the Address Database

The contents of the address database can be dumped using Ethernet frames that get transmitted to the CPU (or other device). Up to 48 valid entries can be retrieved in each frame. See Remote Management described in [Section 8](#).

Alternatively, the contents of the address database can be dumped or searched using the register interface. The dump operation is called Get Next since it returns the active contents of address database in ascending network byte order. A search operation can also be done using the Get Next operation. If multiple address databases are being used (see [Section 2.4.8](#)), set the FID field in the ATU FID register to the database number to search when using the Get Next function.

The Get Next operation starts with the MAC address contained in the ATU MAC registers and returns the next higher active MAC address currently in the address database. Begin with an ATU MAC address of all ones to get the first or lowest active MAC address. The returned MAC address and its associated data is accessible in the ATU MAC and the ATU Data registers. To get the next higher active MAC address, the Get Next operation can be started again without setting the ATU MAC registers since they already contain the 'last' address. A returned ATU MAC address of all ones indicates that no higher active MAC addresses were found or that the Broadcast MAC address was found. In either case, the end of the database has been reached. If it were reached with a valid Broadcast address the entry's EntryState is returned with a non-zero value. A summary of how the Get Next operation uses the ATU's registers is shown in [Table 37](#).

Table 37: ATU Get Next Operation Register Usage

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

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

7.3.3

Loading & Purging an Entry in the Address Database

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

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



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

Note



Static addresses can be modified without their first being purged.

Note

If the address is not found, and if the ATU Data register's EntryState does not equal zero, the address is loaded into the address database using the same protocol as automatic Address Learning (see [Section 2.4.3](#)). The 16 bits of the ATU Data register are written into bits 63:48 of the ATU entry (see [Section 7.3.1](#)). The 3 MACPri bits of the ATU Operation register are written into bits 66:64 of the ATU entry.

A summary of how the Load operation uses the ATU's registers is shown in [Table 38](#).

Table 38: ATU Load/Purge Operation Register Usage

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



Link Street®

88E6321/88E6320 Functional Specification

7.3.4

Flushing Entries

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

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

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

7.3.5

Moving or Removing Single Port Mappings

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

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

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

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

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

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



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

Note



Note

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

7.3.6

Servicing ATU Violations

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

- An ATU Full violation occurs if an Automatic Address Learn ([Section 2.4.3](#)) or an ATU load operation ([Section 7.3.3](#)) could not enter the new MAC address into the address database owing to all four bins at the MAC address's hashed address being locked as static entries.
- An SA Membership Violation occurs when a frame's SA is found in the address database as static and the entry's Destination Port Vector (DPV) data does not align with port's number or mode¹. It also occurs if the port is Locked (Port offset 0x0B), and if the port's RefreshLocked bit is cleared to zero (Port offset 0x0B) and the ATU Age Interrupt is not enabled (Global 2 offset 0x5) and the port's DPV data does not align with port's number or mode. If the ATU Age Interrupt is enabled and the entry's EntryState is less than 0x4 an ATU Miss Violation will be generated instead of the Member Violation. This Membership Violation interrupt can be masked on a per-port basis by setting the port's IgnoreWrongData bit to a one (Port offset 0x0B).
- An SA Miss Violation occurs when a frame's SA is not found in the address database and the port is locked due to the port's LockedPort bit being set to a one (Port offset 0x0B). This 1st learn interrupt can be masked on a per-port basis by clearing the port's LockedPort bit to zero. The SA Miss Violation will also occur if the port is locked and the frame's SA is found in the address database but its EntryState is less than 0x4 and the ATUAgeIntEn bit is set (Global 2 offset 0x5). This refresh interrupt allows the CPU to reload aging ATU entries before they age out, if they are still being used by ingressing frames. The aging interrupt will not happen if the port is configured to auto refresh already learned entries (see RefreshLocked, Port offset 0x0B).
- An ATU Age Out Violation occurs when an entry is at EntryState 0x1 and it is being aged again, and the port the ATU entry is associated with² has its IntOnAgeOut bit set (Port offset 0x0B). Up to two Age Out Violations can be stored and they are serviced ahead of any Full, Member or Miss Violations. With the default AgeTime of 0x16 (330 second age time – Global 1 offset 0x0A) the fastest rate two back-to-back ATU entries can age is 5.75 mSec. So the two deep FIFO will not miss any Age Out Violations unless the CPU takes more than 10 mSec to service the interrupt. For CPU directed learning, it can assure that no Age Out Violations are missed by setting the port's HoldAt1 bit (Port offset 0x0B). The HoldAt1 option ensures that the automatic aging unit never actually purges the entry by decrementing the entry's EntryState to 0x0. Instead it will hold all entries associated with the port at an EntryState of 0x1. The CPU will get another Age Out Violation from this entry on the next age sweep through the address database. When the port's HoldAt1 bit is set, the CPU has to purge all entries as they will not age out on their own.

Captured ATU Violations and their associated interrupts are cleared by the Get/Clear Violation Data ATU Operation. This ATU Operation returns the type of the violation in the ATU Operation register (Global 1 offset 0x0B), the source port that caused the violation in the EntryState/SPID field of the ATU Data register³ (Global 1 offset 0x0C) and returns the MAC address that caused the violation in

1. If the port is not a Trunk port (Port offset 0x05) then the port's bit must be set in the DPV and the entry must not be a Trunk entry (i.e., its 'T' bit must not be set – [Section 7.3.1](#)) or a violation will be generated. If the port is a Trunk port then the entry must be a Trunk entry with its DPV[3:0] matching the port's Trunk ID (Port offset 0x05) or a violation will be generated.
2. An entry is associated with a port when the entry is a non-trunk entry (the entry's 'T' bit = 0) and the port's bit is set in the entry's DPV. If the found entry contains a Trunk ID, the Trunk ID is converted to a DPV using the Trunk Mapping Table (Global 2, offset 0x08) and then it is processed in the same way.
3. Except for ATU Age Out Violations



Link Street®

88E6321/88E6320 Functional Specification

the ATUByte[5:0] fields of the ATU MAC register (Global 1 offsets 0x0D to 0x0F) and the FID that was associated with the frame (Global 1 offset 0x01).

A summary of how the Get/Clear Violation Data operation uses the ATU's registers is shown in Table 39.

Table 39: ATU Get/Clear Operation Register Usage

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

7.3.7 ATU Statistics

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

The address database's 8,192 entries are organized as 2,048 buckets with each bucket containing 4 bins (Section 2.4.1). The buckets are addressed by hashing the 48-bit MAC address down to 11 bits. As multiple MAC addresses can hash to the same bucket (a hash collision), 4 bins in each bucket are provided. The 1st bin is filled first, then the 2nd, and so on.

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

The ATU Statistics are gathered every time an ATU GetNext is performed (Section 7.3.2) and the results are held in the counters until the start of the next GetNext operation. The statistics to gather can be selected prior to the start of the ATU GetNext operation. The choices are:

- Count all valid entries
- Count all valid non-static entries (counts dynamic entries only)
- Count all valid entries in a defined FID (Section 2.4.8)
- Count all valid non-static entries in a defined FID

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

The procedure to collect ATU Statistics is as follows:

1. Set the kind of statistics to collect in the ATU Stats register (Global 2 offset 0x0E).

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

8

Remote Management

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

The benefits of Remote Management are:

- Faster CPU access to large databases of information – specifically the address database (i.e., the ATU – [Section 7.3](#)) and the port statistics (i.e., the MIBs – [Section 2.3.8](#) and [Section 2.3.9](#)).
- Uses the CPU's frame interface (e.g., MII or GMII) to access switch registers saving the need for an alternate interface and its pins on the CPU device.
- Remote access to all switch registers without the need of a local CPU.
- Can control which physical port accepts and processes Remote Management requests (for security).
- Request/Response type protocol. The CPU asks for data only when it wants data. Other methods can be intrusive by cramming information into the CPU at potentially inopportune times.

The Remote Management Unit (RMU) is initially disabled at reset. It needs to be enabled by a CPU using the SMI interface or by an EEPROM attached to the device. The RMU is enabled by setting the RMEnable bit to a one (Global 1 offset 0x1C) and by selecting one of the RMU ports to accept and process RMU frames (using the RMU Mode bits in Global 1 offset 0x1C). Additionally the RMU can be configured to ignore all RMU frames it receives unless the frame's Destination Address (DA) is loaded as static in the address database ([Section 7.3.1](#)). This option is enabled by setting DA Check to a one (Global 1 offset 0x1C).

The RMU can only be enabled on one physical port at a time for security reasons. It needs to be enabled on the port that is directly or indirectly connected to the switch management CPU. The port must also be in either DSA Tag mode or Ether type DSA Tag mode ([Section 5](#)).

**Note**

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

8.1 Request for Frame Format - Layer 2 and DSA Portion

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

The layer 2 portion of the Remote Management frame contains the normal IEEE 802.3 fields of DA, SA, etc. The DA of these frames is defined to be the Marvell® multicast address of 01:50:43:00:00:00¹ or the unicast address of the switch². The SA is the MAC address of the source device. The Distributed Switch Architecture (DSA) portion of the request frame is defined in [Figure 44](#) and the Ether Type (Length/Type) portion of the frame is user definable³. The DSA portion may optionally be Ether Typed DSA Tagged⁴ ([Section 5.9](#)).

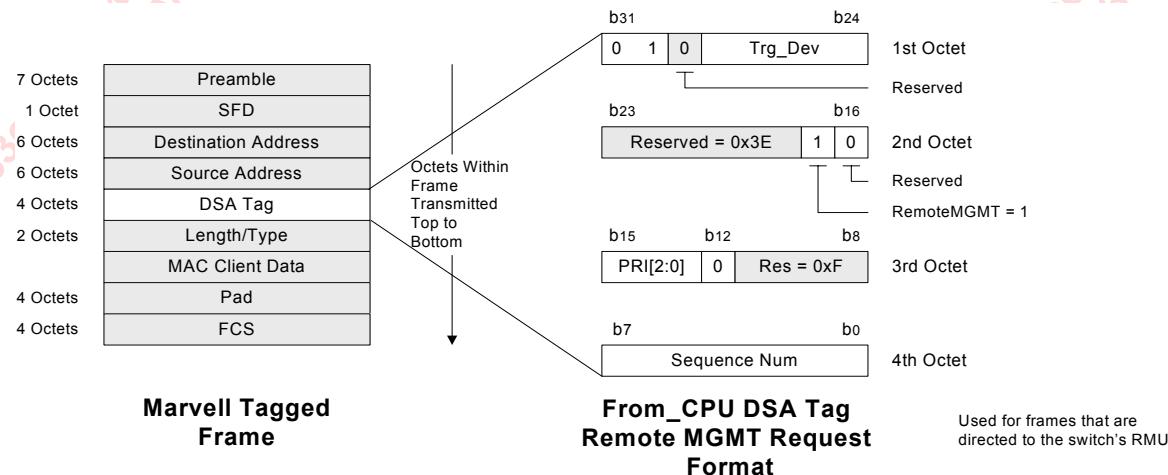
A From_CPU type DSA Tag frame format is used as these frames contain a Trg_Dev field that all DSA enabled devices support that is used to get these frames to the desired switch device if more than one of these devices are interconnected in a box using DSA enabled links. Bit 17 being a one, along with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



All Reserved bits must be the values defined in [Figure 44](#). This means that bits 29, 18, 16 & 12 must be zero and bits 23:19, 17 & 11:8 must be one in the frame.

Note

Figure 44: Remote Management DSA Tag Request Format



Older devices that do not understand this frame format will simply map these frames to the Trg_Dev (and hopefully to a device that does understand the Remote Management function). If Trg_Dev is

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



Link Street®

88E6321/88E6320 Functional Specification

this device, it will process the frame if all the other DSA Tag bits are the correct value, and then the frame will be discarded.

Using the From_CPU DSA Tag gets the Remote Management Request frames to the correct device. This works the same way for DSA and EtherType DSA ([Section 5](#)).

The usage of the DSA Tag's PRI and Sequence Num is discussed in [Section 8.2](#).

8.1.1

RMU and Ether type DSA

Supporting the Ether typed DSA frame format ([Section 5.9](#)) can be used for remote control of a switch using a single link, something a ISP would do to communicate with a remote switch on the outside of someone's house. This application requires the same 'pipe' to be used for both customer data and these Remote Management frames. Security can be assured by allowing Ether typed DSA frames from the provider port only and never from the customer's port(s). For additional security the device supports processing of Remote Management frames only if they ingress a defined physical port (Port 4 or Port 5). This requires that the 'defined' port for Remote Management be either directly or indirectly connected to the management CPU's port. An indirect connection is one where a Remote Management frame hops through one or more other DSA enabled devices before it gets to this device. And the port it enters on this device is the port that is enabled for Remote Management frame processing.

8.1.2

RMU and Marvell® Header

Remote Management is supported on a port where DSA or Ether type DSA frame mode is enabled ([Section 5](#)). It is also supported on a local CPU's port where DSA or Ether type DSA frame mode is enabled together with the Marvell Header Mode being enabled ([Section 6.7](#)). The Marvell Header mode is used to accelerate routing by aligning the layer 3 portion of the Ethernet frame onto a 32-bit boundary. This is done by inserting 2 bytes before the frame's DA.

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

Refer to [Section 8.2.1](#) for restrictions with the use of the Marvell Header mode with Remote Management.

8.2

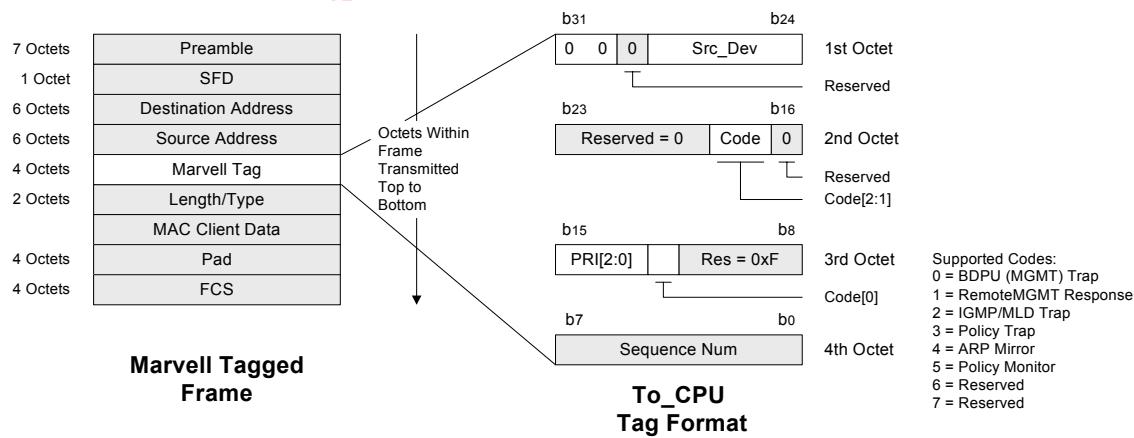
Response Frame Format - Layer 2 and DSA Portion

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

The layer 2 portion of the Remote Management frame contains the normal IEEE 802.3 fields of DA, SA, etc. The DA of these frames is defined to be the SA from the Request frame. The DSA portion of the frame is defined in [Figure 45](#) and the Ether Type (Length/Type) is a copy of the Length/Type field of the Request frame ([Figure 44](#)).

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

Figure 45: Remote Management DSA Tag Response Format



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

The Sequence Num extracted from the Request frame ([Figure 44](#)) is used as the Sequence Num in the Response frame. It is a way for the CPU to match up Responses to Requests in case a Request or a Response gets dropped.

8.2.1

Restrictions of Remote Management

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

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



Link Street®

88E6321/88E6320 Functional Specification

(and subsequent) Request frames will be dropped until the 1st Request has been completely transmitted.

In the device, a separate 512 byte SRAM is used that can be MUX'ed in or out Port 4, Port 5, or Port 6 (register selectable – RMU Mode bits in Global 1 offset 0x1C). The 512 byte SRAM yields a maximum Response Data size of 484 bytes.

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

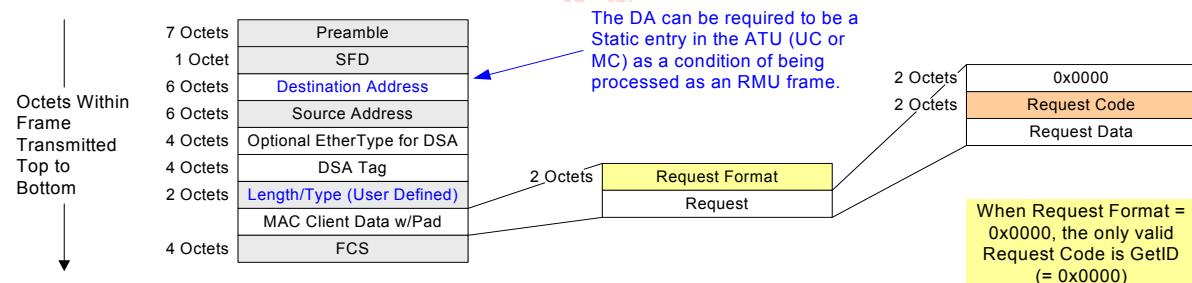
Software will need to look for the From_CPU RemoteMGMT Response code two bytes ahead of where they normally would be if the Header was added to the egressing frame. Since CPU NIC's should not remove the Header when it is there, the full content of these frames will still make it into the CPU's memory.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

8.3

Request Frame Format - Layer 3

The layer 3 portion of the Remote Management request frame contains the specific register action requests. These requests are encapsulated behind a Request Format (Figure 46). The Request Format supports various device families that require very different command and/or response formats due to the internal nature of the specific device family.

Figure 46: Remote Management Generic Layer 3 Request Format



8.3.1

The Initial Request Frame - GetID

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

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

One of the fields returned in the response to a GetID (a GotID – [Section 8.4.1](#)) is the Request Format. This device uses a Request Format of 0x0001 for all other commands.

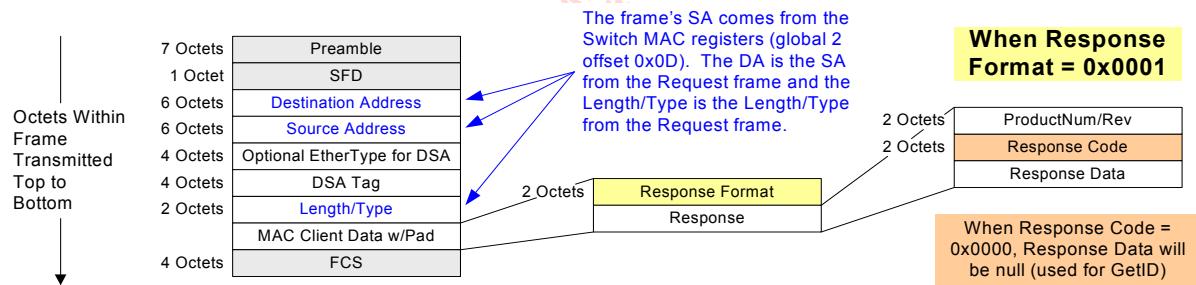
The Request Format is intended to be a Family code where devices that contain the same register interface structure will all be of the same Family. All Marvell® SOHO devices have a similar register interface structure and so they all use the same Request Format even though there are major differences between the SOHO devices. These product specific differences are indicated by the Product-Num/Rev field returned in all Response frames (included in the GotID response to the GetID request). The Response Format is described in [Section 8.4](#).

8.4

Response Frame Format - Layer 3

The layer 3 portion of the Remote Management response frame contains the requested data. These responses are encapsulated behind a Response Format (Figure 47). The Response Format supports various device families that may need very different command and/or response formats due to the internal nature of the specific device family.

Figure 47: Remote Management Generic Layer 3 Response Format



8.4.1

The Initial Response Frame - GotID

The response to the initial request frame of GetID (Section 8.3.1) is a GotID response. A GotID response contains the device's Response Format number followed by the Product Number and Revision of the device, followed by the Response Code and Response Data.

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

8.4.2

Error Handling

The layer 3 portions of the Request frame's Request Code are fully decoded and all frames (that make it into the Remote Management processing unit) are acknowledged with a Response frame. If the CPU sends in a Request Code that is undefined, the hardware will respond with an Error Response frame (Section 8.5.6). This way the software will be able to determine that it got back data it did not expect.

8.5

Supported Requests and Responses

This device uses a Request and Response Format = 0x0001¹.

Non-destructive Remote Management requests can be done at the same time as register operations received on the device's SMI interface pins (MDC and MDIO). Likewise, if the SMI interface limits its actions to non-destructive actions then any Remote Management request can be done at the same time. A destructive action is any request that changes register data. Register writes fall into this category, but so do flush commands (like on the ATU, MIBs, etc.) as well as any register read that causes a bit to be cleared due to the read. Destructive Remote Management commands are noted.

The Request Codes supported in the device are:

- 0x0000 – GetID, [Section 8.5.1](#)
- 0x1000 – Dump ATU, [Section 8.5.2](#)
- 0x1020 – Dump MIBs, [Section 8.5.3](#) (dump only) & [Section 8.5.4](#) (dump & clear)
- 0x2000 – Read/Write Register, [Section 8.5.5](#)

8.5.1

GetID (non-destructive)

The purpose of this request is to get the Response Format and Product/Num Rev of the selected device.

Request Format = 0x0000 <- NOTE: this is 0x0000 for GetID only

Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x0000 (GetID)

Request Data = 0x0000

Response Format = 0x0001 (SOHO)

ProductNum/Rev = [Port 0, offset 0x03]

Response Code = 0x0000 (GotID)

Response Data = null

1. The GetID Request uses a Request Format = 0x0000 but its GotID Response uses the Response Format of the device (i.e., = 0x0001).

8.5.2

Dump ATU (non-destructive)

This request dumps up to 48 valid MAC entries found in the ATU. To start at the beginning of the ATU's memory use a Continue Code = 0x0000. If the ATU is empty the EntryState in the 1st Entry Data field will be 0x0 (bits 7:5 of the 1st octet). If less than 48 valid MAC entries are found, the 1st Entry with an EntryState = 0x0 is the end of the list. If more than 48 valid MAC entries exist in the ATU all 48 entries will be returned with a non-zero EntryState and a (two octet) Continue Code will appear after the 48th Entry Data. The next valid MAC entries can be retrieved by doing a Dump ATU using the Continue Code returned from the previous Dump ATU.

The ATU is dumped in linear ATU memory address order (i.e., they will not be in the GetNext's ascending network byte order) and only valid entries are dumped.

Request Format = 0x0001 (SOHO)

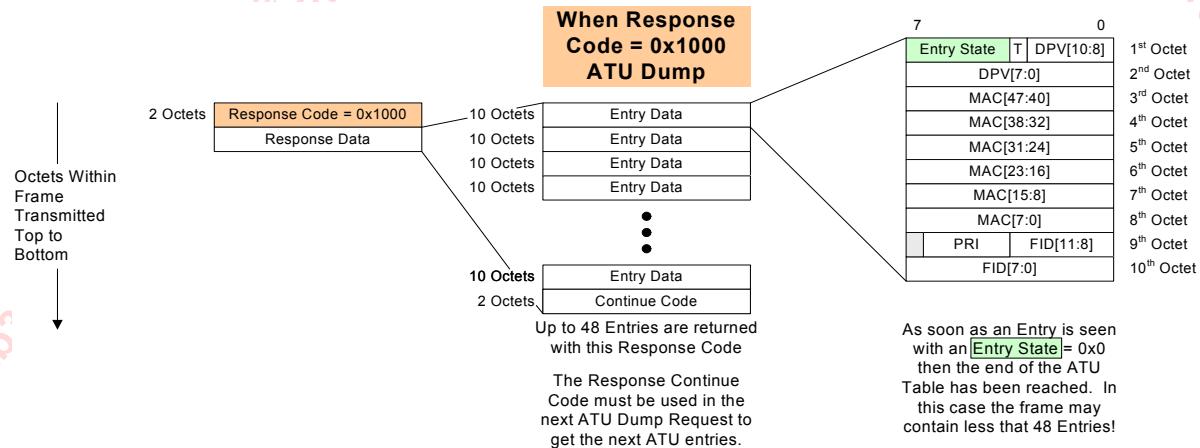
Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x1000 (ATU Dump)

Request Data = [Continue Code] size of 2 Octets (0x0000 = start at beginning of ATU table)

Response = see [Figure 48](#).

Figure 48: Remote Management ATU Dump Response Format



This ATU dump can be occurring in parallel to an ATU GetNext operation received from the SMI pins. This is not recommended to be done for more than the occasional¹ GetNext however, as the GetNext will slow down the ATU Dump's response time.

- Note**
1. An occasional GetNext can be useful as a 'Search' function for a specific MAC address in the ATU. This approach will be faster than dumping the entire ATU contents with multiple frames (and the Search using the SMI pins can be done in parallel with an ATU Dump using a Remote Management frame).

8.5.3

Dump MIBs (non-destructive)

This request dumps all the current MIB counters for a single physical port on the selected device. The physical port number to dump is passed in the Request Data field defined below.

Request Format = 0x0001 (SOHO)

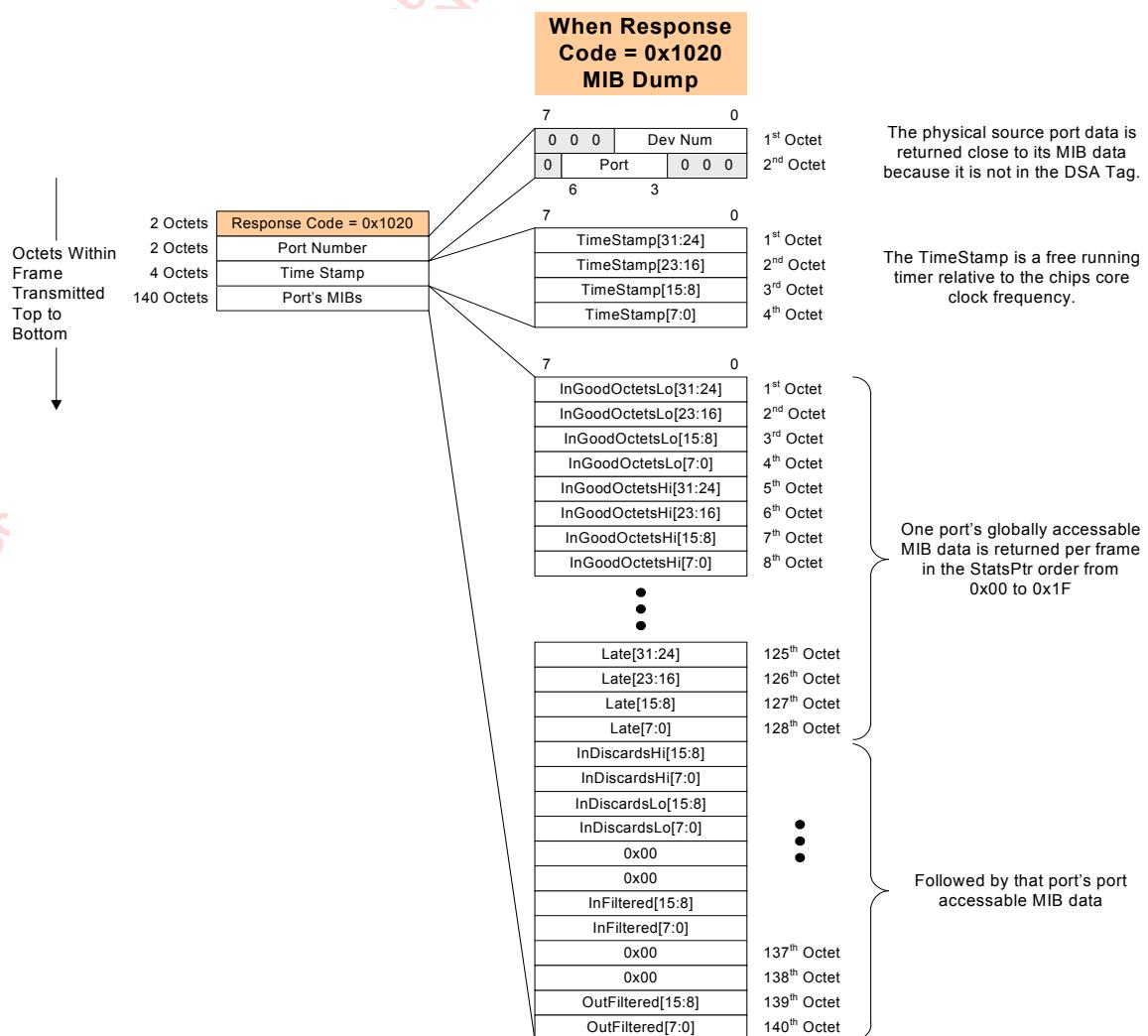
Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x1020 (MIB Dump)

Request Data = 0x000p (p = physical port number) In this device 0x0 => p <= 0xA

Response = see [Figure 49](#).

Figure 49: Remote Management MIB Dump Response Format





Link Street®

88E6321/88E6320 Functional Specification

The TimeStamp is inserted into the frame at the instant the MIB values are just starting to be transferred into the frame. The delta time between two TimeStamp values can be used for rate calculations on the other MIB data.

In this device the TimeStamp increments at a rate of once every 512ns. At this rate the TimeStamp will overflow in about 2,200 seconds or over 36 minutes.



Note The MIBs can be read with this Remote Management request in parallel with the SMI interface capturing and reading the MIBs.

8.5.4 Dump MIBs and Clear (destructive)

This request dumps all the current MIB counters for a single physical port on the selected device and then clears them. This way each Dump MIB & Clear will return the MIB count deltas from the last Dump MIB & Clear request for the same port. The physical port number to dump is passed in the Request Data field defined below.

Request Format = 0x0001 (SOHO)

Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x1020 (MIB Dump)

Request Data = 0x800p (p = physical port number – bit 15 = 1 indicates Clear after Read)

In this device 0x0 => p <= 0xA

Response = see [Figure 49](#).



Note This command is destructive due to the 'Clear' function and should not be done in parallel with the SMI interface accessing the MIBs. Choose one interface or the other

8.5.5 Read/Write Register (may be destructive)

This request reads or writes a single or multiple switch core or PHY registers.



Caution If this command is used to issue a SWReset a Response frame will NOT be generated as the MACs and data path are being reset. Other bit settings can be just as problematic, such as forcing in the Link down or setting the port's PortState to Disabled on the port being used to receive and transmit Remote Management frames.

Request Format = 0x0001 (SOHO)

Pad = 0x0000 (reserved space for reply of ProductNum/Rev)

Request Code = 0x2000 (Register read or write)

Followed by n sets of 4 Octet Register Commands where 0 < n < 121.

Three types of Register Commands are supported:

1. Direct Register read or write

This command is used to read or write 16 bits to any device register as follows:

- 1st Octet [7:4] = 0x0
- 1st Octet [3:2] = Op Code where 10=Read and 01=Write
- 1st Octet [1:0] + 2nd Octet [7:5] = SMI Device Address
- 2nd Octet [4:0] = SMI Register Offset
- 3rd Octet = Data [15:8] (must = 0x00 for Reads)
- 4th Octet = Data [7:0] (must = 0x00 for Reads)

2. Wait on a Bit Command

This command is used to delay processing the next Register Command in the frame until the bit selected is at the desired value. This is typically used to wait until a register's Busy bit clears (like in the ATU or MIBs, etc.). The format of the Wait on a Bit Command is as follows:

- 1st Octet [7:4] = 0x1
- 1st Octet [3:2] = Op Code where 00=Wait until bit is a 0 & 11=Wait until bit is a 1
- 1st Octet [1:0] + 2nd Octet [7:5] = SMI Device Address
- 2nd Octet [4:0] = SMI Register Offset
- 3rd Octet [7:4] = 0x0
- 3rd Octet [3:0] = Bit in register to wait on (0x0 = bit 0, 0x1 = bit 1, ..., 0xF = bit 15)
- 4th Octet = 0x00

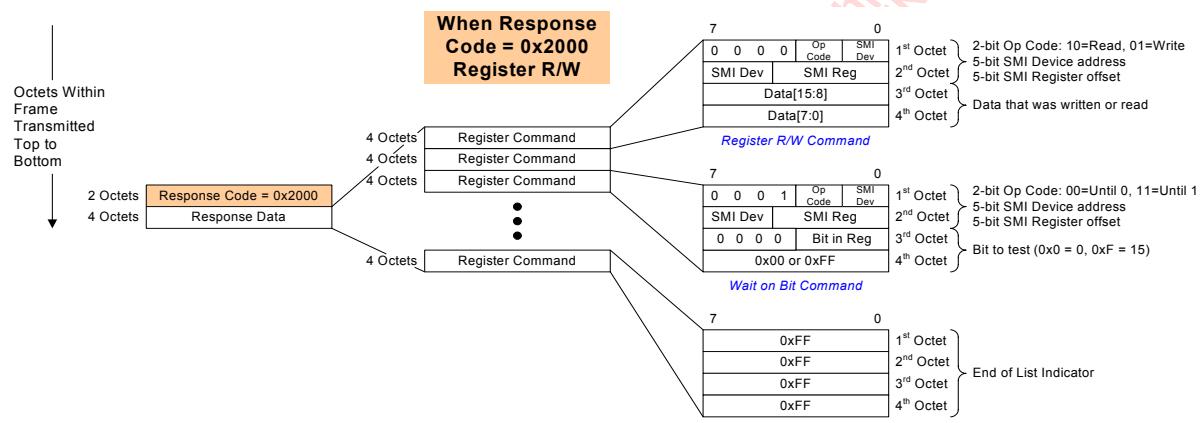
3. End of List Command

This command is used to indicate no more commands occur in the frame. If a frame has the full 121 commands then the 121st must be an End of List. The format of the Wait on a Bit Command is as follows:

- 1st Octet = 0xFF
- 2nd Octet = 0xFF
- 3rd Octet = 0xFF
- 4th Octet = 0xFF

Response = see Figure 50.

Figure 50: Remote Management Register Read/Write Response Format





Link Street®

88E6321/88E6320 Functional Specification

The Response Frame's Response Data is the Request Frames' Request Data with the Data [15:0] portions of Register Read commands being the data that was read from the selected register (i.e., the 0x0000 data that was in the Request frame is replaced with the data read from the register).

Any illegal register format is considered an End of List Command.

If a Wait on Bit command waits for a bit to be in the requested state for more than 1 second, the Wait on Bit command will terminate, return a value of 0xFF in the 4th Octet, and consider this instruction an End of List Indicator (i.e., any commands that follow the failed Wait on Bit command will not be processed).

Reading 'clear on read' registers is considered destructive and should not be done in parallel with destructive SMI operations.

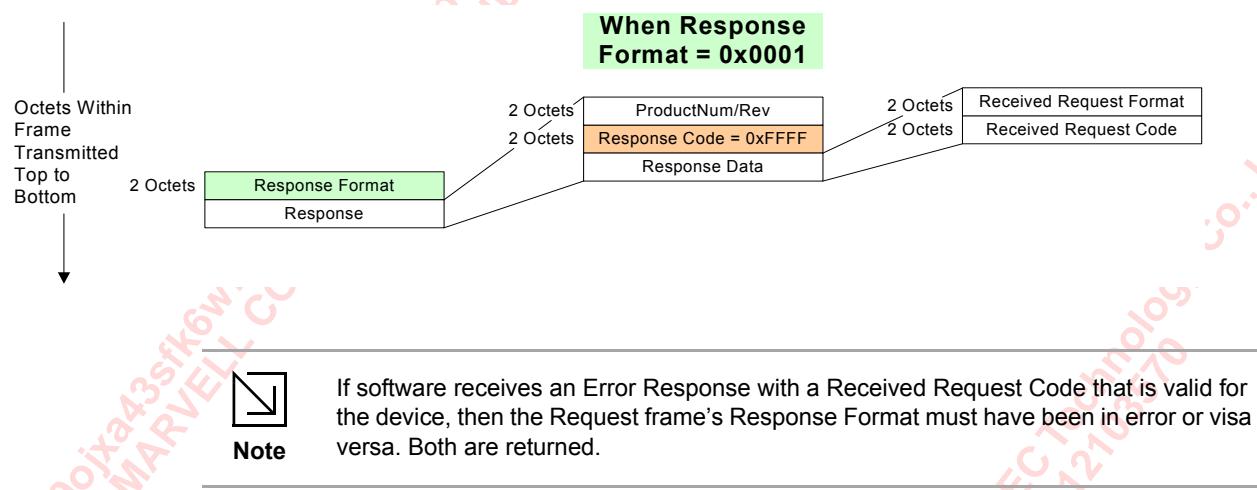
Writing registers is also considered destructive. Only one interface (the Remote Management interface or the SMI interface) should be used to perform destructive operations (i.e., actions that cause changes inside the device).

8.5.6 Error Response Frame (non-destructive)

Any Request frame that contains an unsupported Request Format or Request Code is responded to with the Error Response frame (as ALL requests return a response).

- Request Format = 0x[unsupported] (i.e., Request Format was not 0x0000 or 0x0001 for a SOHO device)
- Pad = 0x0000 (reserved space for reply of ProductNum/Rev)
- Request Code = 0x[undefined] (i.e., the Request Code is an invalid one)
- Request Data = 0x[don't care]
- Response = See [Figure 51](#).

Figure 51: Remote Management Error Response Format



Access to PHY registers is done through the SMI PHY Data and SMI PHY Command registers. Attempts to access the PHY registers directly are invalid but will not generate an Error Response frame. Instead, the write data will not be written and the read data will be returned with 0xFFFF.

Illegal Register Commands will not generate an Error Response frame. They will be interpreted as an End of List command.

An Error Response frame will not be generated if more than 121 Register Commands occur in a Register R/W Request frame. Only the first 121 Register Commands will be processed and then an automatic End of List command will be assumed.



Link Street®

88E6321/88E6320 Functional Specification

9

PHY and SERDES Functional Specifications

9.1

PHY Functional Description

The 88E6321/88E6320 device includes two 10/100/1000BASE-T Gigabit Ethernet transceivers with integrated termination resistors (Port 3 and Port 4). The following sections describe the PHYs in the device. The PHY registers are defined in [Section 11.1, PHY Register Description, on page 443](#).

9.1.1

Media Interface

The copper interface consists of the Px_MDIP/N[3:0] signals that connect to the physical media for 1000BASE-T, 100BASE-TX, and 10BASE-T modes of operation.

The device integrates MDI interface termination resistors. The IEEE 802.3 specification requires that both sides of a link have termination resistors to prevent reflections. Traditionally, these resistors and additional capacitors are placed on the board between a PHY device and the magnetics. The resistors have to be very accurate to meet the strict IEEE return loss requirements. Typically, $\pm 1\%$ accuracy resistors are used on the board. These additional components between the PHY and the magnetics complicate board layout. Integrating the resistors has many advantages including component cost savings, better ICT yield, board reliability improvements, board area savings, improved layout, and signal integrity improvements.

9.1.1.1

Transmit Side Network Interface

Multi-mode TX Digital to Analog Converter

The device incorporates a multi-mode transmit DAC to generate filtered 4D PAM 5, MLT3, or Manchester coded symbols. The transmit DAC performs signal wave shaping to reduce EMI. The transmit DAC is designed for very low parasitic loading capacitances to improve the return loss requirement, which allows the use of low cost transformers.

Slew Rate Control and Waveshaping

In 1000BASE-T mode, partial response filtering and slew rate control is used to minimize high frequency EMI. In 100BASE-TX mode, slew rate control is used to minimize high frequency EMI. In 10BASE-T mode, the output waveform is pre-equalized via a digital filter.

9.1.1.2

Encoder

1000BASE-T

In 1000BASE-T mode, the transmit data bytes are scrambled to 9-bit symbols and encoded into 4D PAM5 symbols. Upon initialization, the initial scrambling seed is determined by the PHY address. This prevents multiple device from outputting the same sequence during idle, which helps to reduce EMI.

100BASE-TX

In 100BASE-TX mode, the transmit data stream is 4B/5B encoded, serialized, and scrambled.

10BASE-T

In 10BASE-T mode, the transmit data is serialized and converted to Manchester encoding.

9.1.1.3 Receive Side Network Interface

Analog to Digital Converter

The device incorporates an advanced high speed ADC on each receive channel with greater resolution than the ADC used in the reference model of the IEEE 802.3ab standard committee. Higher resolution ADC results in better SNR, and therefore, lower error rates. Patented architectures and design techniques result in high differential and integral linearity, high power supply noise rejection, and low metastability error rate. The ADC samples the input signal at 125 MHz.

Active Hybrid

The device employs a sophisticated on-chip hybrid to substantially reduce the near-end echo, which is the super-imposed transmit signal on the receive signal. The hybrid minimizes the echo to reduce the precision requirement of the digital echo canceller. The on-chip hybrid allows both the transmitter and receiver to use the same transformer for coupling to the twisted pair cable, which reduces the cost of the overall system.

Echo Canceller

Residual echo not removed by the hybrid and echo due to patch cord impedance mismatch, patch panel discontinuity, and variations in cable impedance along the twisted pair cable result in drastic SNR degradation on the receive signal. The device employs a fully developed digital echo canceller to adjust for echo impairments from more than 100 meters of cable. The echo canceller is fully adaptive to compensate for the time varying nature of channel conditions.

NEXT Canceller

The 1000BASE-T physical layer uses all 4 pairs of wires to transmit data to reduce the baud rate requirement to only 125 MHz. This results in significant high frequency crosstalk between adjacent pairs of cable in the same bundle. The device employs 3 parallel NEXT cancellers on each receive channel to cancel any high frequency crosstalk induced by the adjacent 3 transmitters. A fully adaptive digital filter is used to compensate for the time varying nature of channel conditions.

Baseline Wander Canceller

Baseline wander is more problematic in the 1000BASE-T environment than in the traditional 100BASE-TX environment due to the DC baseline shift in both the transmit and receive signals. The device employs an advanced baseline wander cancellation circuit to automatically compensate for this DC shift. It minimizes the effect of DC baseline shift on the overall error rate.

Digital Adaptive Equalizer

The digital adaptive equalizer removes inter-symbol interference at the receiver. The digital adaptive equalizer takes unequalized signals from ADC output and uses a combination of feedforward equalizer (FFE) and decision feedback equalizer (DFE) for the best-optimized signal-to-noise (SNR) ratio.

Digital Phase Lock Loop

In 1000BASE-T mode, the slave transmitter must use the exact receive clock frequency it sees on the receive signal. Any slight long-term frequency phase jitter (frequency drift) on the receive signal must be tracked and duplicated by the slave transmitter; otherwise, the receivers of both the slave and master physical layer devices have difficulty canceling the echo and NEXT components. In the device, an advanced DPLL is used to recover and track the clock timing information from the receive



Link Street®

88E6321/88E6320 Functional Specification

signal. This DPLL has very low long-term phase jitter of its own, thereby maximizing the achievable SNR.

Link Monitor

The link monitor is responsible for determining if link is established with a link partner. In 10BASE-T mode, link monitor function is performed by detecting the presence of valid link pulses (NLPs) on the MDIP/N pins.

In 100BASE-TX and 1000BASE-T modes, link is established by scrambled idles.

If Force Link Good register 16_0.10 is set high, the link is forced to be good and the link monitor is bypassed for 100BASE-TX and 10BASE-T modes. In the 1000BASE-T mode, register 16_0.10 has no effect.

Signal Detection

In 1000BASE-T mode, signal detection is based on whether the local receiver has acquired lock to the incoming data stream.

In 100BASE-TX mode, the signal detection function is based on the receive signal energy detected on the MDIP/N pins that is continuously qualified by the squelch detect circuit, and the local receiver acquiring lock.

9.1.1.4 Decoder

1000BASE-T

In 1000BASE-T mode, the receive idle stream is analyzed so that the scrambler seed, the skew among the 4 pairs, the pair swap order, and the polarity of the pairs can be accounted for. Once calibrated, the 4D PAM 5 symbols are converted to 9-bit symbols that are then descrambled into 8-bit data values. If the descrambler loses lock for any reason, the link is brought down and calibration is restarted after the completion of Auto-Negotiation.

100BASE-TX

In 100BASE-TX mode, the receive data stream is recovered and converted to NRZ. The NRZ stream is descrambled and aligned to the symbol boundaries. The aligned data is then parallelized and 5B/4B decoded. The receiver does not attempt to decode the data stream unless the scrambler is locked. The descrambler “locks” to the *scrambler* state after detecting a sufficient number of consecutive idle code-groups. Once locked, the descrambler continuously monitors the data stream to make sure that it has not lost synchronization. The descrambler is always forced into the *unlocked* state when a link failure condition is detected, or when insufficient idle symbols are detected.

10BASE-T

In 10BASE-T mode, the recovered 10BASE-T signal is decoded from Manchester to NRZ, and then aligned. The alignment is necessary to ensure that the start of frame delimiter (SFD) is aligned to the nibble boundary.

9.1.2 Loopback

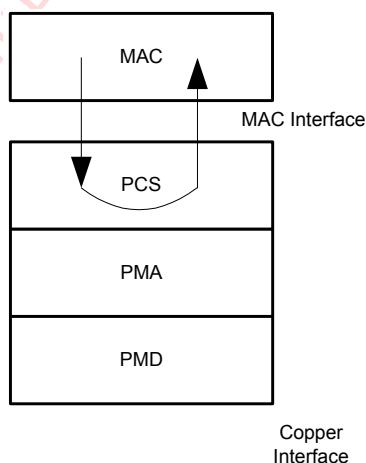
The copper PHY implements various loopbacks.

9.1.2.1 MAC Interface Loopback

The functionality, timing, and signal integrity of the internal MAC interface can be tested by placing the PHY's internal MAC interface loopback mode. This can be accomplished by setting register 0_0.14 = 1. In loopback mode, the data received from the MAC is not transmitted out on the media interface. Instead, the data is looped back and sent to the MAC. During loopback, link will be lost and packets will not be received.

If Auto-Negotiation occurs while loopback is enabled, FLP Auto-Negotiation codes will be transmitted. If in forced 10BASE-T mode and loopback is enabled, 10BASE-T idle link pulses will be transmitted on the copper side. If in forced 100BASE-T mode and loopback is enabled, 100BASE-T idles will be transmitted on the copper side.

Figure 52: MAC Interface Loopback Diagram



The speed of the interface is determined by register 21_2.2:0.

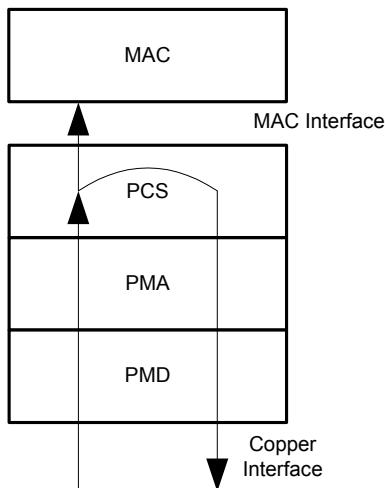
Table 40: MAC Speed Control

Control Register - MAC																	
Register	Function	Setting	Mode	HW Rst	SW Rst												
21_2.2.0	Default MAC interface speed	<p>Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by software reset to take effect.</p> <p>MAC Interface Speed during Link down while Auto-Negotiation is enabled.</p> <table> <tr><td>Bit</td><td>Speed</td></tr> <tr><td>0XX</td><td>Reserved</td></tr> <tr><td>100</td><td>10 Mbps</td></tr> <tr><td>101</td><td>100 Mbps</td></tr> <tr><td>110</td><td>1000 Mbps</td></tr> <tr><td>111</td><td>Reserved</td></tr> </table>	Bit	Speed	0XX	Reserved	100	10 Mbps	101	100 Mbps	110	1000 Mbps	111	Reserved	R/W	110	Update
Bit	Speed																
0XX	Reserved																
100	10 Mbps																
101	100 Mbps																
110	1000 Mbps																
111	Reserved																

9.1.2.2 Line Loopback

Line loopback allows a link partner to send packets into the PHY to test the transmit and receive data path. Packets from a link partner into the PHY, before reaching the MAC interface pins, are looped back and sent out on the line. They are also sent to the MAC. The packets received from the MAC are ignored during line loopback. Refer to [Figure 53](#). This allows the link partner to receive its own packets.

Figure 53: Line Loopback Data Path



Before enabling the line loopback feature, the PHY must first establish link to another PHY link partner. If Auto-Negotiation is enabled, both link partners should advertise the same speed and full-duplex. If Auto-Negotiation is disabled, both link partners need to be forced to the same speed and full-duplex. Once link is established, enable the line loopback mode by writing to register 21_2.14

- 21_2.14 = 1 (Enable line loopback)
- 21_2.14 = 0 (Disable line loopback)

9.1.2.3

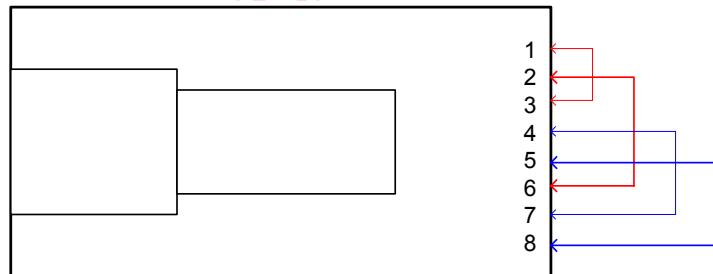
External Loopback

For production testing, an external loopback stub allows testing of the complete data path.

For 10BASE-T and 100BASE-TX modes, the loopback test requires no register writes. For 1000BASE-T mode, register 18_6.3 must be set to 1 to enable the external loopback. All copper modes require an external loopback stub.

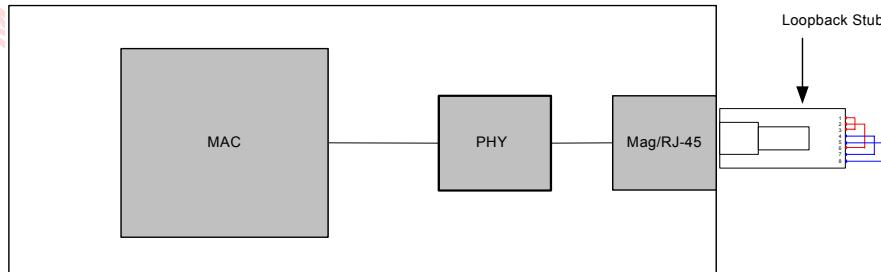
The loopback stub consists of a plastic RJ-45 header, connecting RJ-45 pair 1,2 to pair 3,6 and connecting pair 4,5 to pair 7,8, as seen in [Figure 54](#).

Figure 54: Loopback Stub (Top View with Tab up)



The external loopback test setup requires the presence of a MAC that will originate the packets to be sent out through the PHY. Instead of a normal RJ-45 cable, the loopback stubs allows the PHY to self-link at 10 Mbps, 100 Mbps, or 1000 Mbps. It also allows the actual external loopback. See [Figure 55](#). The MAC should see the same packets it sent, looped back to it.

Figure 55: Test Setup for 10/100/1000 Mbps Modes using an External Loopback Stub





Link Street®

88E6321/88E6320 Functional Specification

9.1.3

Synchronizing FIFO

The integrated PHYs have a transmit synchronizing FIFO to reconcile frequency differences between the clocks of the MAC interface and the media side. The depth of the transmit FIFO can be programmed via register bits 16_2.15:14. See the "Alaska® Ultra FAQs" for details on how to calculate required FIFO depth and the details of the different clocks used for transmit and receive in each mode of operation.

The FIFO depths can be increased in length by programming Register 16_2.15:14 to support longer packets. The deeper the FIFO depth, the higher the latency will be.

The status of the FIFO can be interrogated. Register 19_2.7 indicates the transmit FIFO overflow and underflow status.

Registers 19_2.3 and 19_2.2 are set depending on whether the FIFO inserted or deleted idle symbols. Idles inserted or deleted will be flagged only if the inter packet gap is 24 bytes or less at the input of the FIFO. Inserted or deleted idles will be ignored if the inter packet gap is greater than 24 bytes.

The FIFO status bits can generate interrupts by setting the corresponding bits in register 18_2.

Table 41: FIFO Status Bits

Register	Function	Setting
19_2.7	FIFO Over/Underflow	1 = Over/Underflow error 0 = No FIFO error
19_2.3	FIFO Idle Inserted	1 = Idle inserted 0 = No Idle inserted
19_2.2	FIFO Idle Deleted	1 = Idle deleted 0 = Idle not deleted

9.1.4 Power Management

The PHY supports several advanced power management modes that conserve power.

9.1.4.1 Low Power Modes

Three low power modes are supported in the PHY.

- IEEE 22.2.4.1.5 compliant power down
- Power down with signal detect mode
- Energy Detect (Mode 1)
- Energy Detect+™ (Mode 2)

IEEE 22.2.4.1.5 power down compliance allows for the PHY to be placed in a low-power consumption state by register control.

Power down with signal detect mode is similar to IEEE 22.2.4.1.5 power down, except the signal detect logic remains powered up.

Energy Detect (Mode 1) allows the PHY to wake up when energy is detected on the wire.

Energy Detect+™ (Mode 2) is identical to Mode 1 with the additional capability to wake up a link partner. In Mode 2, 10BASE-T link pulses are sent once every second while listening for energy on the line.

Details of each mode are described below.

9.1.4.2 Low Power Mode Descriptions

IEEE Power Down Mode

The standard IEEE power down mode is entered by setting register 0_0.11 = 1. In this mode, the PHY does not respond to any activity on the copper media.

In this power down mode, the PHY cannot wake up on its own by detecting activity on the media. It can only wake up by setting registers 0_0.11 = 0 and 16_0.2 = 0.

Power Down With Signal Detect Mode

This mode is entered by setting register 0_0.11 = 0 and 16_0.2 = 1. In this mode, the PHY does not respond to any activity on the copper media.

In this power down mode, the PHY cannot wake up on its own by detecting activity on the media. It can only wake up by setting registers 0_0.11 = 0 and 16_0.2 = 0.

Energy Detect Power Down Modes

The PHY can be placed in energy detect power down modes by selecting either of the two energy detect modes. Both modes enable the PHY to wake up on its own by detecting activity on the CAT 5 cable. The status of the energy detect is reported in register 17_0.4 and the energy detect changes are reported in register 19_0.4. **Energy Detect (Mode 1)**

Energy Detect (Mode 1) is entered by setting register 16_0.9:8 to 10.

In Mode 1, only the signal detection circuitry and serial management interface are active. How the PHY wakes up depends on the register setting of 16_0.7. If 16_0.7 is set to 1, the PHY can wake up by writing 16_0.4 to 1. If 16_0.7 is set to 0, the PHY can automatically wake up when energy is detected on the wire. When the PHY wakes up, it starts to Auto-Negotiate sending FLPs for 5 seconds. If at the end of 5 seconds the Auto-Negotiation is not completed, then the PHY stops sending FLPs and goes back to monitoring receive energy. If Auto-Negotiation is completed, then the PHY goes into normal 10/100/1000 Mbps operation. If during normal operation the link is lost,



Link Street®

88E6321/88E6320 Functional Specification

the PHY will re-start Auto-Negotiation. If no energy is detected after 5 seconds, the PHY goes back to monitoring receive energy.

Energy Detect +™ (Mode 2)

Energy Detect (Mode 2) is entered by setting register 16_0.9:8 to 11.

In Mode 2, the PHY sends out a single 10 Mbps NLP (Normal Link Pulse) every one second. Except for this difference, Mode 2 is identical to Mode 1 operation. If the PHY is in Mode 1, it cannot wake up a connected device; therefore, the connected device must be transmitting NLPs, or either device must be woken up through register access. If the PHY is in Mode 2, then it can wake a connected device.

Power State Upon Exiting Power Down

When the PHY exits power down (register 0_0.11 or 16_0.2) the active state will depend on whether the energy detect function is enabled (register 16_0.9:8 = 1x). If the energy detect function is enabled, the PHY will transition to the energy detect state first and will wake up only if there is a signal on the wire.

Table 42: Power State after Exiting Power Down

Register 0_0.11	Register 16_0.2	Register 16_0.9:8	Behavior
1	x	xx	IEEE Power down
0	1	xx	Power down with signal detect
1 to 0	1	xx	Transition to power down with signal detect state
1 to 0	0	00	Transition to power up
0	1 to 0	00	Transition to power up
1 to 0	0	1x	Transition to energy detect state
0	1 to 0	1x	Transition to energy detect state

Normal 10/100/1000 Mbps Operation

Normal 10/100/1000 Mbps operation can be entered by turning off energy detect mode by setting register 16_0.9:8 to 00.

9.1.5

SERDES Manual Power Down

The SERDES power down control bit is described in [Table 43](#). The power down control independently powers down its respective circuits. In general, it is not necessary to power down an unused interface. The PHY will automatically power down any unused circuit. For example when auto-media detect is turned on, the unused interface will automatically power down.

The automatic PHY power management can be overridden by setting the power down control bits. These bits have priority over the PHY power management in that the circuit can not be powered up by the power management when its associated power down bit is set to 1. When a circuit is power back up by setting the bit to 0, a software reset is also automatically sent to the corresponding circuit.

Table 43: SERDES Power Down Control Bits

Reset Register	Register Effect
0_1.11	Fiber/SGMII Power Down



Link Street®

88E6321/88E6320 Functional Specification

9.1.6 Auto-Negotiation

Auto-Negotiation provides a mechanism for transferring information from the local station to the link partner to establish speed, duplex, and Master/Slave preference during a link session.

Auto-Negotiation is initiated upon any of the following conditions:

- Power up reset
- Hardware reset
- Software reset (Register 0_0.15)
- Restart Auto-Negotiation (Register 0_0.9)
- Transition from power down to power up (see [Table 42](#))
- The link goes down

The 10/100/1000BASE-T Auto-Negotiation (AN) is based on Clause 28 and 40 of the IEEE802.3 specification. It is used to negotiate speed, duplex, and flow control over CAT5 UTP cable. Once Auto-Negotiation is initiated, the PHY determines whether or not the remote device has Auto-Negotiation capability. If so, the PHY and the remote device negotiate the speed and duplex with which to operate.

If the remote device does not have Auto-Negotiation capability, the PHY uses the parallel detect function to determine the speed of the remote device for 100BASE-TX and 10BASE-T modes. If link is established based on the parallel detect function, then it is required to establish link at half-duplex mode only. Refer to IEEE 802.3 clauses 28 and 40 for a full description of Auto-Negotiation.

After hardware reset, 10/100/1000BASE-T Auto-Negotiation can be enabled and disabled via Register 0_0.12. Auto MDI/MDIX and Auto-Negotiation may be disabled and enabled independently. When Auto-Negotiation is disabled, the speed and duplex can be set via registers 0_0.13, 0_0.6, and 0_0.8 respectively. When Auto-Negotiation is enabled the abilities that are advertised can be changed via registers 4 and 9.

Changes to registers 0_0.12, 0_0.13, 0_0.6 and 0_0.8 do not take effect unless one of the following takes place:

- Software reset (registers 0_0.15)
- Restart Auto-Negotiation (register 0_0.9)
- Transition from power down to power up (see [Table 42](#))
- The copper link goes down

To enable or disable Auto-Negotiation, Register 0_0.12 should be changed simultaneously with either register 0_0.15 or 0_0.9. For example, to disable Auto-Negotiation and force 10BASE-T half-duplex mode, register 0_0 should be written with 0x8000.

Registers 4_0 and 9_0 are internally latched once every time the Auto-Negotiation enters the Ability Detect state in the arbitration state machine. Hence, a write into Register 4_0 or 9_0 has no effect once the PHY begins to transmit Fast Link Pulses (FLPs). This guarantees that sequences of FLPs transmitted are consistent with one another.

Register 7_0 is treated in a similar way as registers 4_0 and 9_0 during additional next page exchanges.

If 1000BASE-T mode is advertised, then the PHY automatically sends the appropriate next pages to advertise the capability and negotiate Master/Slave mode of operation. If the user does not wish to transmit additional next pages, then the next page bit (Register 4_0.15) can be set to zero, and the user needs to take no further action.

If next pages in addition to the ones required for 1000BASE-T are needed, then the user can set register 4_0.15 to one, and send and receive additional next pages via registers 7_0 and 8_0,

respectively. The PHY stores the previous results from register 8 in internal registers, so that new next pages can overwrite register 8_0.

Note that 1000BASE-T next page exchanges are automatically handled by the PHY without user intervention, regardless of whether or not additional next pages are sent.

Once the PHY completes Auto-Negotiation, it updates the various status in registers 1_0, 5_0, 6_0, and 10_0. Speed, duplex, page received, and Auto-Negotiation completed status are also available in registers 17_0 and 19_0.

See 11 "PHY and SERDES Register Description" on page 443.

9.1.6.1

Reverse Auto-Negotiation

When reverse Auto-Negotiation is enabled the Auto-Negotiation process proceeds as usual except the priority resolution is resolved in the order shown in Table 44. The IEEE registers 4_0.8:5 and 9_0.9:8 are used to advertise the capability just like in normal Auto-Negotiation. Register 26_0.14 controls whether gigabit mode should be advertised regardless of the values in registers 9_0.9:8.

Table 44: Reverse Auto-Negotiation Priority Resolution

Priority	Speed/Duplex
1 (Highest)	10 Full-duplex
2	10 Half-duplex
3	100 Full-duplex
4	100 Half-duplex
5	1000 Full-duplex
6 (Lowest)	1000 Half-duplex

Registers 26_0.14 and 26_0.13 controls the Auto-Negotiation behavior as shown in Table 46.

The enabling of reverse Auto-Negotiation, the disabling of 1000BASE-T, and the enabling of the downshift feature create some conflicts. Table 45 clarifies the behavior.



Link Street®

88E6321/88E6320 Functional Specification

Table 45: Reverse Auto-Negotiation, Disable 1000BASE-T, and Downshift Behavior

Reverse Auto-Negotiation (26_0.13)	Disable 1000BASE-T (26_0.14)	Downshift (16_0.11)	Action
Reverse Auto-Negotiation	Disable 1000BASE-T	Don't care	Will negotiate to the lowest of 10/100
Reverse Auto-Negotiation	Enable 1000BASE-T	Don't care	Will negotiate to the lowest of 10/100/1000
Normal Auto-Negotiation	Disable 1000BASE-T	Don't care	Will negotiate to the highest of 10/100
Normal Auto-Negotiation	Enable 1000BASE-T	Enabled	Will downshift to 10/100 if needed
Normal Auto-Negotiation	Enable 1000BASE-T	Disabled	Will negotiate to the highest of 10/100/1000

The reverse Auto-Negotiation mechanism works by temporarily stalling the base page exchange until the link partners 10/100 capabilities are learned. Once the PHY learns the link partner's capability it advertises only the desired capability, in this case the lowest speed with the highest duplex.

Whenever Auto-Negotiation is restarted all advertised capabilities are advertised to the link partner. If reverse Auto-Negotiation is enabled then the pause bits for every other FLP burst are inverted. Since the link partner never sees 3 consecutive FLP burst having the same bit pattern its Auto-Negotiation is stalled. At the same time the link partner advertises its capabilities. Once the PHY learns the link partner's capabilities it determines the lowest common speed/duplex. It then no longer advertises any higher capabilities in the FLP burst. This new set of advertisement is used and the remainder of the Auto-Negotiation process continues. (i.e., the FLP burst pause bits are no longer toggling so the link partner's Auto-Negotiation process is no longer stalled.)

Note that if 2 PHYs with reverse Auto-Negotiation enabled are connected to each other, the process described above will still work. Since all advertised capabilities are initially sent to the link partner and the speed/duplex bits are not toggling it is possible for each PHY to determine each other's capabilities.

Table 46: Reverse Auto-Negotiation Registers

Register	Function	Setting	Mode	HW Rst	SW Rst
0_0.9	Restart Auto-Negotiation	1 = Restart Auto-Negotiation Process 0 = Normal operation	R/W, SC	0	SC
26_0.14	Disable 1000BASE-T	When set to disabled, 1000BASE-T will not be advertised even if registers 9_0.9 or 9_0.8 are set to 1. A write to this register bit does not take effect until any one of the following occurs: <ul style="list-style-type: none">• Software reset is asserted (Register 0_0.15)• Restart Auto-Negotiation is asserted (Register 0_0.9)• Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation• Copper link goes down. 1 = Disable 1000BASE-T Advertisement 0 = Enable 1000BASE-T Advertisement	R/W	See Descr.	Update
26_0.13	Reverse Auto-neg	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) <ul style="list-style-type: none">• Restart Auto-Negotiation is asserted (Register 0_0.9)• Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation• Copper link goes down. 1 = Reverse Auto-Negotiation 0 = Normal Auto-Negotiation	R/W	See Descr.	Update



Link Street®

88E6321/88E6320 Functional Specification

9.1.7

Downshift Feature

Without the downshift feature enabled, connecting between two Gigabit link partners requires a four-pair RJ-45 cable to establish 10, 100, or 1000 Mbps link. However, there are existing cables that have only two-pairs, which are used to connect 10 Mbps and 100 Mbps Ethernet PHYs. With the availability of only pairs 1, 2 and 3,6, Gigabit link partners can Auto-Negotiate to 1000 Mbps, but fail to link. The Gigabit PHY will repeatedly go through the Auto-Negotiation but fail 1000 Mbps link and never try to link at 10 Mbps or 100 Mbps.

With the Marvell® downshift feature enabled, the PHY is able to Auto-Negotiate with another Gigabit link partner using cable pairs 1,2 and 3,6 to downshift and link at 10 Mbps or 100 Mbps, whichever is the next highest advertised speed common between the two Gigabit PHYs.

In the case of a three pair cable (additional pair 4,5 or 7,8 - but not both) the same downshift function for two-pair cables applies.

By default, the downshift feature is turned off. Refer to register 16_0.14:11 which describe how to enable this feature and how to control the downshift algorithm parameters.

To enable the downshift feature, the following registers must be set:

- Register 16_0.11 = 1 - Enables downshift
- Register 16_0.14:12 - Sets the number of link attempts before downshifting

9.1.8

Fast 1000BASE-T Link Down Indication

Per the IEEE 802.3 Clause 40 standard, a 1000BASE-T PHY is required to wait 750 milliseconds or more to report that link is down after detecting a problem with the link. For Metro Ethernet applications, a Fast Failover in 50 ms is specified, which cannot be met if the PHY follows the 750 ms wait time. This delay can be reduced by intentionally violating the IEEE standard by setting register 26_0.9 to 1.

The delay at which link down is to be reported can be selected by setting register 26_0.11:10. 00 = 0ms, 01 = 10 ± 2 ms, 10 = 20 ± 2 ms, 11 = 40 ± 2 ms.

9.1.9

Advanced Virtual Cable Tester® (VCT™)

The PHY device Virtual Cable Tester feature uses Time Domain Reflectometry (TDR) to determine the quality of the cables, shorts, cable impedance mismatch, bad connectors, termination mismatch, and bad magnetics. The PHY device transmits a signal of known amplitude (+1V) down each of the four pairs of an attached cable. It will conduct the cable diagnostic test on each pair, testing the Px_MDIP/N[0], Px_MDIP/N[1], Px_MDIP/N[2], and Px_MDIP/N[3] pairs sequentially. The transmitted signal will continue down the cable until it reflects off of a cable imperfection.

The Advanced VCT™ has 4 modes of operation that are programmable via register 23_5.7:6. The first mode returns the peak with the maximum amplitude that is above a certain threshold. The second mode returns the first peak detected that is above a certain threshold. The third mode measures the systematic offset at the receiver. The fourth mode measures the amplitude seen at a certain specified distance.

The VCT test is initiated by setting register 23_5.15 to 1. This bit will self clear when the test is completed. Register 23_5.14 will be set to a 1 indicating that the TDR results in the registers are valid.

Each point in the VCT reflected waveform is sampled multiple times and averaged. The number of samples to take is programmable via register 23_5.10:8.

Each time the VCT test is enabled, the results seen on the four receive channels are reported in registers 16_5, 17_5, 18_5, and 19_5. Register 23_5.13:11 selects which channel transmits the test pulse.

When register 23_5.13:11 is set to 000 the same channel reflection is recorded. In other words, channel 0 transmits a pulse and the reflection seen on channel 0 receiver is reported. Channel 1 transmits a pulse and the reflection seen on channel 1 receiver is reported. The same for channel 2 and channel 3.

When register 23_5.13:11 is set to 100 all 4 receive channels report the reflection seen by a pulse transmitted by channel 0.

When register 23_5.13:11 is set to 101 all 4 receive channels report the reflection seen by a pulse transmitted by channel 1.

When register 23_5.13:11 is set to 110 all 4 receive channels report the reflection seen by a pulse transmitted by channel 2.

When register 23_5.13:11 is set to 111 all 4 receive channels report the reflection seen by a pulse transmitted by channel 3.

Hence, if only the reflection seen on the same channel is desired the VCT test should be run with 23_5.13:11 = 000. If all same channel and cross channel combinations are desired then the VCT test must be run 4 times with 23_5.13:11 set to 100, 101, 110, and 111 for the 4 runs. Registers 16_5, 17_5, 18_5, and 19_5 should be read and stored between each run.

9.1.9.1

Maximum Peak

When register 23_5.7:6 is set to 00, the maximum peak above a certain threshold is reported. Pulses are sent out and recorded according to the setting of register 25_5.13:11.

There are 5 threshold settings for same channel reflections and they are specified by registers 26_5.6:0, 26_5.14:8, 27_5.6:0, 27_5.14:8, and 28_5.6:0. These settings correspond to the amplitude threshold the reflected signal has to exceed before it is counted. Any reflected signal below this threshold level is ignored. The 5 threshold settings are based on cable length with the breakpoints at 10 m, 50 m, 110 m, and 140 m.

There are 2 threshold settings for cross-channel specified by registers 25_5.6:0 and 25_5.14:8 and are based on cable length with the breakpoint at 10 m.

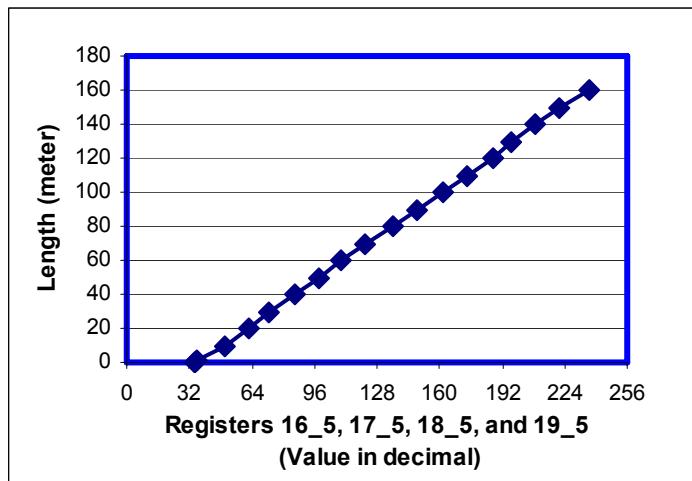
The default values are targeted to 85 ohm to 115 ohms. However these threshold settings should be calibrated for the desired impedance setting on the target system.

The results are stored in registers 16_5, 17_5, 18_5, and 19_5. Bits 7:0 reports the distance of the peak. The distance can be converted to using the trend line in Figure 56. Bits 14:8 reports the reflected amplitude. Bit 15 reports whether the reflected amplitude was positive or negative. When bit 15:8 returns a value of 0x80 it means there was no peak detected above the threshold. If bit 15:8 returns a value of 0x00 then the test failed.

Register 28_5.7 controls the exact distance that is reported. When set to 0 the distance where the amplitude falls to 50% of the peak amplitude is reported. When set to 1 the distance where the peak amplitude actually occurs is reported. In either case, the magnitude of the maximum amplitude is reported in bits 14:8.

In the maximum peak mode register 24_5.7:0 is used to set the starting distance of the sweep. Normally this value should be set to 0. If this value is set to a non-zero value, any reflection below the starting distance is ignored. Note that 24_5.9:8 are ignored. The maximum peak only measures up to about 200 meters of cable.

Figure 56: Cable Fault Distance Trend Line



9.1.9.2 First Peak

When register 23_5.7:6 is set to 01, the first peak above a certain threshold is reported. The first peak operates in exactly the same way as the maximum peak except that there has to be some qualification as to what constitute a peak since the first peak is not necessarily the maximum peak. The first peak is defined as the maximum amplitude seen before the reflected amplitude drops by some value below this peak. This hysteresis value is defined by register 23_5.5:0.

For example in Figure 57 if Pa is greater than the hysteresis value in 23_5.5:0 and Va is above the threshold value, then Va and A are reported since it is the first peak that is above the threshold.

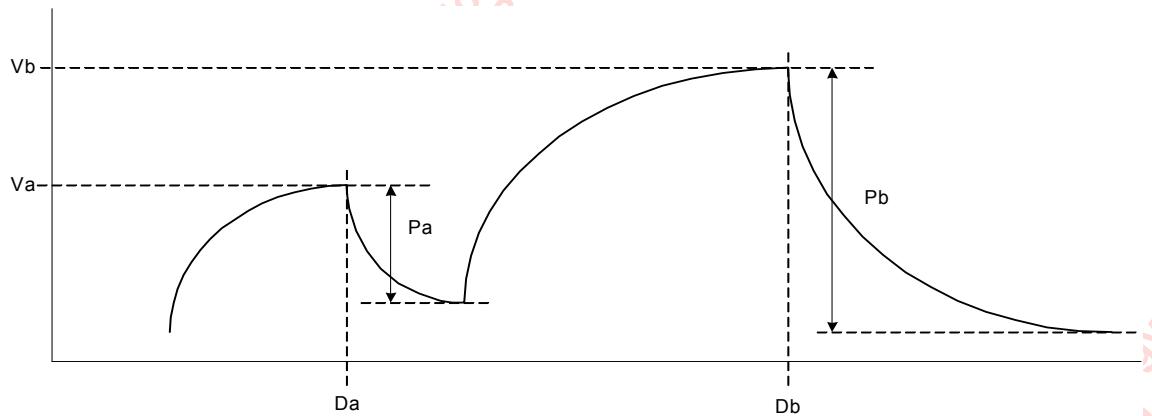
If Pa is less than the hysteresis value in 23_5.5:0 then Va and Da are not reported as the first peak. Vb and Db will be reported as the first peak if Pb is greater than the hysteresis value in 23_5.5:0 and Vb is above the threshold value.

If Pa is greater than the hysteresis value in 23_5.5:0 but Va is below the threshold value then Va and Da are not reported as the first peak. Vb and Db will be reported as the first peak if Pb is greater than the hysteresis value in 23_5.5:0 and Vb is above the threshold value.

Register 28_5.7 controls the exact distance that is reported. When set to 0 the distance where the amplitude falls below the peak amplitude minus the hysteresis level as defined in register 23_5.5:0 is reported. When set to 1 the distance where the peak amplitude actually occurs is reported. In either case, the magnitude of the maximum amplitude of the first peak is reported in bits 14:8.

In the first peak mode register 24_5.7:0 is used to set the starting distance of the sweep. Normally this value should be set to 0. If this value is set to a non-zero value, any reflection below the starting distance is ignored. This may be useful to ignore reflections at the transformer to be reported as the first peak. Note that 24_5.9:8 are ignored. The maximum peak only measures up to about 200 meters of cable.

Figure 57: First Peak Example



9.1.9.3 Offset

The offset, indicated by 23_5.7:6 = 10, reports the offset seen at the receiver. This is more of a debug mode. Bits 7:0 of registers 16_5, 17_5, 18_5, and 19_5 have no meaning. When bit 15:8 returns a value of 0x80 it means there is zero offset. If bit 15:8 returns a value of 0x00 then the test failed.

Note that in the maximum peak, first peak, and sample point modes, the systematic offset is automatically subtracted from the results.

9.1.9.4 Sample Point

When register 23_5.7:6 is set to 11, the amplitude of the reflected pulse at a particular distance on the cable is reported. Unlike the maximum peak and first peak modes which only measures up to about 200 meters of cable, the sample point mode can measure up to 400 meters of cable.

The sample point returns the amplitude of the reflected pulse at a particular distance on the cable. The distance is set by register 24_5.9:0. The threshold registers 25_5, 26_5, 27_5, and 28_5.6:0 are ignored.

Bits 7:0 of registers 16_5, 17_5, 18_5, and 19_5 returns the same value as 24_5.7:0. Note that registers 24_5.9 is not returned. Bits 14:8 returns the amplitude, and bit 15 the sign of the amplitude. If the test failed bits 15:8 will return 00000000 (zero amplitude will always return as 10000000).

By programming register 24_5.9:0 from 0x000 to 0x1FF and running the sample point test at each distance it is possible to reconstruct the reflected amplitude. Note that since the threshold is ignored, it is possible that some small reflections in the same channel measurements will be reported at short cable lengths when there are none. This is because the analog hybrid does not 100% cancel out the transmitted signal.



Link Street®

88E6321/88E6320 Functional Specification

9.1.9.5**Pulse Amplitude and Pulse Width**

The transmitted pulse amplitude and pulse width can be adjusted via registers 28_5.9:8 and 28_5.11:10 respectively. They should normally be set to full amplitude and 1/4 pulse width. See [Table 361, Advanced VCT™ Same Pair Impedance Positive Threshold 4 and Transmit Pulse Control, on page 481](#) for details.

9.1.9.6**Drop Link**

When register 28_5.12 is set to 0 the circuit will wait 1.5 seconds to break the link before starting VCT™. When set to 1 this delay is bypassed.

9.1.9.7**VCT™ with Link Up**

The following status requires the PHY to link up with a link partner.

Register 20_5 reports the pair skew of each pair of wire relative to each other.

Register 21_5.3:0 reports the polarity of each pair of wire.

Register 21_5.5:4 reports the crossover status

Register 20_5 and 21_5 are not valid unless register 21_5.6 is set to 1.

9.1.10**Data Terminal Equipment (DTE) Detect**

The PHY supports the Data Terminal Equipment (DTE) power function. The DTE power function is used to detect if a link partner requires power supplied by the PHY.

The DTE power function can be enabled by writing to register 26_0.8, followed by a software reset. When DTE is enabled, the PHY will first monitor for any activity transmitted by the link partner. If the link partner is active, then the link partner has power and power from the PHY is not required. If there is no activity coming from the link partner, DTE power engages, and special pulses are sent to detect if the link partner requires DTE power. If the link partner has a low pass filter (or similar fixture) installed, it will be detected that the link partner requires DTE power.

The DTE power status register (Register 17_0.2) immediately comes up as soon as it is detected that a device (link partner) requires DTE power. Register 19_0.2 indicates when DTE power status has changed states.

If the power supplied to a link partner that requires DTE power is disconnected, the DTE power status (register 17_0.2) will drop after a user-controlled delay. The user-controlled delay can be programmed using Register 26_0.7:4 (default is 20 seconds). This delay is needed to ensure the link partner is powered up and the low-pass filter (or similar fixture) is removed before the DTE power status is evaluated and reported. If desired, the DTE power status drop can be reported immediately, by writing register 26_0.7:4 to 4'b0000.

A detailed description of the register bits used for DTE power detection for the PHY are shown in [Table 47](#).

Table 47: Registers for DTE Power

Register	Description
26_0.8 - Enable power over Ethernet detection	1 = Enable DTE detect 0 = Disable DTE detect
17_0.2 - Power over Ethernet detection status	1 = Link partner needs DTE power 0 = Link partner does not need DTE power
19_0.2 - Power over Ethernet detection state changed	1 = DTE power detection status changed 0 = No DTE power detection status change detected
26_0.7:4 - DTE detect status drop	0000: report immediately 0001: report 5s after DTE power status drop ... 1111: report 75s after DTE power status drop



Link Street®

88E6321/88E6320 Functional Specification

9.1.11

Energy Efficient Ethernet (EEE)

The device implements Energy Efficient Ethernet (EEE) functions based on IEEE 802.3az. The device supports EEE on the following System and Media Interfaces:

- 10BASE-Te
- 100BASE-T EEE
- 1000BASE-T EEE

9.1.11.1

Energy Efficient Ethernet (EEE) Auto-Negotiation

In order for the EEE to work, both the local PHY and its link partner need to support EEE. The 100/1000BASE-T EEE capabilities are exchanged through standard 10/100/1000BASE-T Auto-Negotiation (IEEE 802.3 clauses 28 and 40). These capabilities are exchanged through the Next Page exchanges. The Auto-Negotiation process is used only for the exchange of speed and duplex information and not any timer information with regards to the sleep, refresh, and wake cycles.

10BASE-Te capability is not part of 10/100/1000BASE-T Auto-Negotiation. 10BASE-Te can be enabled by writing Reg 20_0.7 = '1'. When the 10BASE-Te is enabled, the copper media transmit waveform is changed to save some power consumption.

9.1.12

CRC Error Counter and Packet Counter

The CRC error counter and packet counters, normally found in MACs, are available in the PHY. The error counter and packet counter features are enabled through register writes and each counter is stored in eight register bits.

9.1.12.1

Enabling The CRC Error Counter and Packet Counter

To enable both counters to count, set 16_6.4 to 1.

To disable and clear both counters, set 16_6.4 to 0.

To read the CRC error counter and packet counter, read register 17_6.

17_6.15:8 (Packet count is stored in these bits)

17_6.7:0 (CRC error count is stored in these bits)

The counter does not clear on a read command. To clear the CRC error counter, disable and enable the counters.

9.1.13

Packet Generator

The PHY contains a very simple packet generator. Link should be established first prior to enabling the packet generator. The generator will generate packets at the speed of the established link.

Once enabled, fixed length packets of 64 or 1518 bytes (including CRC) will be transmitted continuously separated by 12 bytes of IPG. The preamble length will be 8 bytes. The payload of the packet is either a fixed 5A, A5, 5A, A5 pattern or a pseudo random pattern. A correct IEEE CRC is appended to the end of the packet. An error packet can also be generated.

The registers are as follows:

16_6.3 Packet generation enable. 0 = Normal operation, 1 = Enable internal packet generator

16_6.2 Payload type. 0 = Pseudo random, 1 = Fixed 5A, A5, 5A, A5,...

16_6.1 Packet length. 0 = 64 bytes, 1 = 1518 bytes

16_6.0 Error packet. 0 = Good CRC, 1 = Symbol error and corrupt CRC.

16_6.15:8 Packet Burst Size. 0x00 = Continuous, 0x01 to 0xFF = Burst 1 to 255 packets.

If register 16_6.15:8 is set to a non-zero value, then register 16_6.3 will self clear once the required number of packets are generated. Note that if register 16_6.3 is manually set to 0 while packets are still bursting, the bursting will cease immediately once the current active packet finishes transmitting. The value in register 16_6.15:8 should not be changed while 16_6.3 is set to 1.



Link Street®

88E6321/88E6320 Functional Specification

9.1.14 MDI/MDIX Crossover

The PHY automatically determines whether or not it needs to cross over between pairs as shown in [Table 48](#) so that an external crossover cable is not required. If the PHY interoperates with a device that cannot automatically correct for crossover, the PHY makes the necessary adjustment prior to commencing Auto-Negotiation. If the PHY interoperates with a device that implements MDI/MDIX crossover, a random algorithm as described in IEEE 802.3 clause 40.4.4 determines which device performs the crossover.

When the PHY interoperates with legacy 10BASE-T devices that do not implement Auto-Negotiation, the PHY follows the same algorithm as described above since link pulses are present. However, when interoperating with legacy 100BASE-TX devices that do not implement Auto-Negotiation (i.e. link pulses are not present), the PHY uses signal detect to determine whether or not to crossover.

The auto MDI/MDIX crossover function can be disabled via register 16_0.6:5.

The pin mapping in MDI and MDIX modes is shown in [Table 48](#).

Table 48: Media Dependent Interface Pin Mapping

Pin	MDI			MDIX		
	1000BASE-T	100BASE-TX	10BASE-T	1000BASE-T	100BASE-TX	10BASE-T
MDIP/N[0]	BI_DA±	TX±	TX±	BI_DB±	RX±	RX±
MDIP/N[1]	BI_DB±	RX±	RX±	BI_DA±	TX±	TX±
MDIP/N[2]	BI_DC±	unused	unused	BI_DD±	unused	unused
MDIP/N[3]	BI_DD±	unused	unused	BI_DC±	unused	unused



[Table 48](#) assumes no crossover on PCB.

Note

The MDI/MDIX status is indicated by Register 17_0.6. This bit indicates whether the receive pairs (3,6) and (1,2) are crossed over. In 1000BASE-T operation, the PHY can correct for crossover between pairs (4,5) and (7,8) as shown in the table above. However, this is not indicated by Register 17_0.6.

If 1000BASE-T link is established, pairs (1,2) and (3,6) crossover is reported in register 21_5.4, and pairs (4,5) and (7,8) crossover is reported in register 21_5.5.

9.1.15

Polarity Correction

The PHY automatically corrects polarity errors on the receive pairs in 1000BASE-T and 10BASE-T modes. In 100BASE-TX mode, the polarity does not matter.

In 1000BASE-T mode, receive polarity errors are automatically corrected based on the sequence of idle symbols. Once the descrambler is locked, the polarity is also locked on all pairs. The polarity becomes unlocked only when the receiver loses lock.

In 10BASE-T mode, polarity errors are corrected based on the detection of validly spaced link pulses. The detection begins during the MDI crossover detection phase and locks when the 10BASE-T link is up. The polarity becomes unlocked when link is down.

The polarity correction status is indicated by Register 17_0.1. This bit indicates whether the receive pair (3,6) is polarity reversed in MDI mode of operation. In MDIX mode of operation, the receive pair is (1,2) and Register 17_0.1 indicates whether this pair is polarity reversed. Although all pairs are corrected for receive polarity reversal, Register 17_0.1 only indicates polarity reversal on the pairs described above.

If 1000BASE-T link is established register 21_5.3:0 reports the polarity on all 4 pairs.

Polarity correction can be disabled by register write 16_0.1 = 1. Polarity will then be forced in normal 10BASE-T mode.

9.1.16

FLP Exchange Complete with No Link

Sometimes when link does not come up, it is difficult to determine whether the failure is due to the Auto-Negotiation FLP exchange not completing or from the 10/100/1000BASE-T link not able to come up.

Register 19_0.3 is a sticky bit that gets set to 1 whenever the FLP exchange is completed but the link cannot be established for some reason. Once the bit is set, it can be cleared only by reading the register.

This bit will not be set if the FLP exchange is not completed, or if link is established.



Link Street®

88E6321/88E6320 Functional Specification

9.1.17 Interrupt

Registers 18_0 and 18_2 are the Interrupt Enable registers for the copper media and the interface, respectively.

Registers 19_0 and 19_2 are the Interrupt Status registers for the copper media and the interface, respectively.

Register 18_0 or 18_2 determine whether a interrupt pin is asserted when an interrupt event occurs. Register 19_0 or 19_2 reports interrupt status. When an interrupt event occurs, the corresponding bit in register 19_0 or 19_2 are set and remains set until register 19_0 or 19_2 are read via the serial management interface. When interrupt enable bits are not set in register 18_0 or 18_2, interrupt status bits in register 19_0 or 19_2 are still set when the corresponding interrupt events occur. However, an interrupt pin is not asserted.

An interrupt pin is asserted as long as one interrupt status bit is set in register 19_0 or 19_2 with its corresponding interrupt enable bit set in register 18_0 or 18_2.

To de-assert interrupt pins

- Clear register 19_0 or 19_2 via a management read
- Disable the interrupt enable by writing register 18_0 or 18_2

9.1.18 Temperature Sensor

The device contains an internal temperature sensor. Register 26_6.4:0 reports the die temperature and is updated approximately once per second. The result can be read back on any port as long as the port is not disabled (i.e. register 0.11 = 1).

An interrupt can be generated when the temperature exceeds a certain threshold.

Register 26_6.6 is set high whenever the temperature is greater than or equal to the value programmed in register 26_6.12:8. Register 26_6.6 remains high until read.

Register 26_6.7 controls whether the interrupt pin is asserted when register 26_6.6 is high.

The interrupt should be enabled on only one port since there is only 1 temperature sensor for the entire chip.

Table 49: Misc Test Register (Temperature Sensor Register 1)

Register	Function	Setting	Mode	HW Rst	SW Rst
26_6.12:8	Temperature Threshold	Temperature in C = 5 x 26_6.4:0 - 25 i.e., for 100C the value is 11001	R/W	11001	Retain
26_6.7	Temperature Sensor Interrupt Enable	1= Interrupt Enable 0 = Interrupt Enable	R/W	0	Retain
26_6.6	Temperature Sensor Interrupt	1 = Temperature Reached Threshold 0 = Temperature Below Threshold	RO, LH	0	0
26_6.4:0	Temperature Sensor (5-bit)	Temperature in C = 5 x 26_6.4:0 - 25 i.e., for 100C the value is 11001	RO	xxxxx	xxxxx

Register 27_6.7:0 provides an alternate temperature sensor reading in 1C increments. The result can be read back on any port as long as the port is not disabled. The sampling rate and number of samples to average settings can be changed by writing to Reg 27_6.12:11 and Reg 27_6.10:8 respectively.

Table 50: Temperature Sensor Register (Temperature Sensor Register 2)

Register	Function	Setting	Mode	HW Rst	SW Rst
27_6.15:13	Reserved		R/W	0	Retain
27_6.12:11	Temperature Sensor Number of Samples to Average	00 = Average over 5*2^9 samples 01 = Average over 5*2^11 samples 10 = Average over 5*2^13 samples 11 = Average over 5*2^15 samples	R/W	01	Retain
27_6.10:8	Temperature Sensor Sampling Rate	Sampling Rate 000 = Reserved 001 = Reserved 010 = 168 us 011 = 280 us 100 = 816 us 101 = 2.28 ms 110 = 6.22 ms 111 = 11.79 ms	R/W	100	Retain
27_6.7:0	Temperature Sensor Alternative Reading (8-bit)	Temperature in C = 1 x 27_6.7:0 - 25. i.e. for 100C the value is 0111_1101	RO	x	x

9.2

SERDES Functional Description

The 88E6321/88E6320 device includes two Gigabit SERDES interfaces that support SGMII, 1000BASE-X and 100BASE-FX. The following sections describe the Gigabit SERDES in the device. The SERDES registers are defined in [Section 11.2, SERDES Register Description, on page 492](#).

9.3

1.25 GHz SERDES Interface

The 1.25 GHz SERDES Interface can be configured as an SGMII to be hooked up to a MAC or as a 100BASE-FX/1000BASE-X/SGMII to be hooked up to the media.

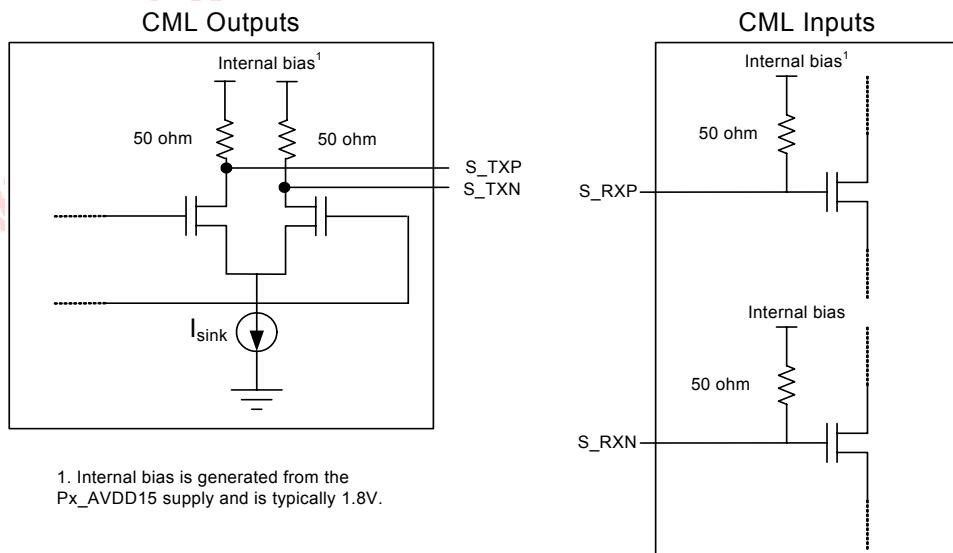
9.3.1

Electrical Interface

The input and output buffers are internally terminated to 50 ohm impedance. The output swing can be adjusted by programming register 26_1.2:0.

The input and output buffers of the 1.25 GHz SERDES interface are internally terminated by 50 ohm impedance. No external terminations are required. The 1.25 GHz SERDES I/Os are Current Mode Logic (CML) buffers. CML I/Os can be used to connect to other components with PECL or LVDS I/Os. See the “Reference Design Schematics” and “Fiber Interface” application note for details.

Figure 58: CML I/Os



9.3.2

SGMII Speed and Link

Two registers are available to determine whether the SGMII achieved link and sync. Register 17_1.5 indicates that the SERDES locked onto the incoming KDKDKD... sequence. Register 17_1.10 indicates whether 1000BASE-X link is established on the SERDES. If SGMII Auto-Negotiation is disabled, register 17_1.10 has the same meaning as register 17_1.5. If SGMII Auto-Negotiation is enabled, then register 17_1.10 indicates whether SGMII Auto-Negotiation successfully established link.

9.3.3

SGMII TRR Blocking

When the SGMII receives a packet with odd number of bytes, a single symbol of carrier extension will be passed on and transmitted onto 1000BASE-T. This carrier extension may cause problems with full-duplex MACs that incorrectly handle the carrier extension symbols. When register 16_1.13 is set to 1, all carrier extend and carrier extend with error symbols received by the SGMII will be converted to idle symbols when operating in full-duplex. Carrier extend and carrier extend with error symbols will not be blocked when operating in half-duplex, or if register 16_1.13 is set to 0. Note that symbol errors will continue to be propagated regardless of the setting of register 16_1.13.

This function is on by default as the SGMII rev 1.8 standard requires this function to be implemented.

9.3.4

False SERDES Link Up Prevention

The SERDES interface can operate in 1000BASE-X mode and in 100BASE-FX mode where an unconnected optical receiver will send full swing noise into the PHY. Sometimes this random noise will look like a real signal and falsely cause the 1000BASE-X or 100BASE-FX PCS to link up.

A noise filtering state machine can be enabled to reduce the probability of false link up. When the state machine is enabled it will cause a small delay in link up time.

Table 51: Fiber Noise Filtering

Register	Function	Setting	Mode	HW Rst	SW Rst
26_1.14	1000BASE-X Noise Filtering	1 = Enable 0 = Disable	R/W	0	Retain
26_1.13	100BASE-FX Noise Filtering	1 = Enable 0 = Disable	R/W	0	Retain

9.4 Loopback

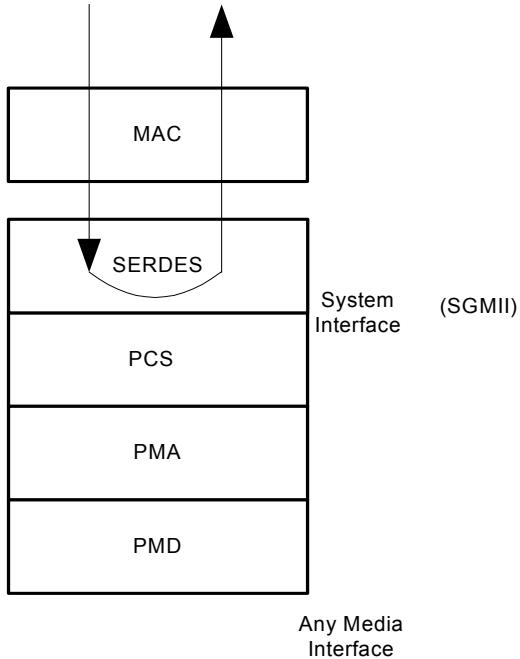
The device implements various different loopback paths.

9.4.1 Synchronous SERDES Loopback

The 1.25 GHz SERDES can loop back the raw 10 bit symbol at the receiver back to the transmitter. No frequency compensation is performed when the 10 bit symbol is looped back. This mode facilitates testing using non 8/10 symbols such as PRBS.

The 1.25 GHz SERDES synchronous loopback is enabled by setting register 16_1.12 = 1 and 16_1.8 = 1.

Figure 59: Synchronous SERDES Loopback Diagram



9.5 Resets

In addition to the hardware reset pin (RESETn) there are several software reset bits as summarized in [Table 52](#).

The entire chip is reset as if the RESETn pin is asserted. Once triggered, registers are not accessible through the MDIO until the chip reset completes.

The copper and fiber circuits are reset per port via register 0_0.15, 0_1.15, and 0_4.15 respectively. A reset in one circuit does not directly affect another circuit.

All the reset registers described are self clear.

Table 52: Reset Control Bits

Reset Register	Register Effect	Block
0_1.15	Software Reset for Bank 1	Fiber/SGMII - per port



Link Street®

88E6321/88E6320 Functional Specification

9.6

Auto-Negotiation

The device supports four types of Auto-Negotiation.

- 1000BASE-X Fiber Auto-Negotiation (IEEE 802.3 Clause 37)
- SGMII Auto-Negotiation (Cisco specification)

Auto-Negotiation provides a mechanism for transferring information from the local station to the link partner to establish speed, duplex, and Master/Slave preference during a link session.

Auto-Negotiation is initiated upon any of the following conditions:

- Power up reset
- Hardware reset
- Software reset (Register 0_0.15, 0_1.15, or 0_4.15)
- Restart Auto-Negotiation (Register 0_0.9, 0_1.9, 0_4.9)
- Transition from power down to power up (Register 0.0_0.11, 0_1.11, or 0_4.11)
- The link goes down

The following sections describe each of the Auto-Negotiation modes in detail.

9.6.1

1000BASE-X Auto-Negotiation

1000BASE-X Auto-Negotiation is defined in Clause 37 of the IEEE 802.3 specification. It is used to auto-negotiate duplex and flow control over fiber cable. Registers 0_1, 4_1, 5_1, 6_1, and 15_1 are used to enable AN, advertise capabilities, determine link partner's capabilities, show AN status, and show the duplex mode of operation respectively.

Fiber Register 22.7:0 must be set to one to view the fiber auto-negotiation registers.

The device supports Next Page option for 1000BASE-X Auto-Negotiation. Register 7_1 of the fiber pages is used to transmit Next Pages, and register 8_1 of the fiber pages is used to store the received Next Pages. The Next Page exchange occurs with software intervention. The user must set Register 4_1.15 to enable fiber Next Page exchange. Each Next Page received in the registers should be read before a new Next Page to be transmitted is loaded in Register 7_1.

If the PHY enables 1000BASE-X Auto-Negotiation and the link partner does not, the link cannot link up. The device implements an Auto-Negotiation bypass mode. See [Section 9.6.2.1](#) for more details.

9.6.2

SGMII Auto-Negotiation

SGMII is a de-facto standard designed by Cisco. SGMII uses 1000BASE-X coding to send data as well as Auto-Negotiation information between the PHY and the MAC. However, the contents of the SGMII Auto-Negotiation are different than the 1000BASE-X Auto-Negotiation. See the "Cisco SGMII Specification" and the "MAC Interfaces and Auto-Negotiation" application note for further details.

9.6.2.1

Serial Interface Auto-Negotiation Bypass Mode

If the MAC or the PHY implements the Auto-Negotiation function and the other does not, two-way communication is not possible unless Auto-Negotiation is manually disabled and both sides are configured to work in the same operational modes. To solve this problem, the device implements the SGMII Interface Auto-Negotiation Bypass Mode. When entering the state "Ability_Detect", a bypass timer begins to count down from an initial value of approximately 200 ms. If the device receives idles during the 200 ms, the device will interpret that the other side is "alive" but cannot send configuration codes to perform Auto-Negotiation. After 200 ms, the state machine will move to a new state called "Bypass_Link_Up" in which the device assumes a link-up status and the operational mode is set to the value listed under the "Comments" column of [Table 53](#).

Table 53: SGMII Auto-Negotiation modes

Reg. 0_1.12	Reg. 26_1.6	Comments
0	X	No Auto-Negotiation. User responsible for determining speed, link, and duplex status by reading PHY registers.
1	0	Normal SGMII Auto-Negotiation. Speed, link, and duplex status automatically communicated to the MAC during Auto-Negotiation.
1	1	MAC Auto-Negotiation enabled. Normal operation. MAC Auto-Negotiation disabled. After 200 ms the PHY will disable Auto-Negotiation and link based on idles.



Link Street®

88E6321/88E6320 Functional Specification

9.7

1.25G PRBS Generator and Checker

A PRBS generator and checker are available for use on the 1.25G SERDES. PRBS7 is supported.

A 32-bit checker is implemented. Note that the reads are atomic. A read to the LSB will update the MSB register. The counters only clear when register 23_1.4 is set to 1. This bit self clears.

The checker and generator polarity can be inverted by setting registers 23_1.7 and 23_1.6 respectively.

Register 23_1.5 controls whether the checker has to lock before counting commences.

Table 54: 1.25 GHz SERDES PRBS Registers

Register	Function	Setting
23_1.7	Invert Checker Polarity	0 = Invert 1 = Normal
23_1.6	Invert Generator Polarity	0 = Invert 1 = Normal
23_1.5	PRBS Lock	0 = Counter Free Runs 1 = Do not start counting until PRBS locks first
23_1.4	Clear Counter	0 = Normal 1 = Clear Counter
23_1.1	PRBS Checker Enable	0 = Disable 1 = Enable
23_1.0	PRBS Generator Enable	0 = Disable 1 = Enable
24_1.15:0	PRBS Error Count LSB	A read to this register freezes register 25_1. Cleared only when register 23_1.4 is set to 1.
25_1.15:0	PRBS Error Count MSB	This register does not update unless register 24_1 is read first. Cleared only when register 23_1.4 is set to 1.

9.8

Unidirectional Transmit

IEEE 802.3ah requires OAM support with unidirectional transmit capability. Unidirectional transmit allows a PHY to transmit data when the PHY does not have link due to potential issues on the receive path. IEEE 802.3ah formally requires two bits for this capability. Register 0.5 enables this capability, and 1.7 advertises this ability. This ability only applies to 100BASE-TX or 1000BASE-X. It doesn't apply to 1000BASE-T since 1000BASE-T requires MASTER/SLAVE relationship and training with both link partners participating, which requires that link exists for any data transmit.

The device can support transmit of packets when there is no link by using register bit 16_0.10 = 1 (Force Copper Link Good) and 16_1.10 = 1(Force Fiber Link Good). This is not the official bit specified by the IEEE 802.3ah but serves the same function.

9.9

Synchronous Clocking Support (88E6321 Device Only)

Synchronous Ethernet (Sync-E) is used to get the entire system to run frequency locked with no frequency drift. One port's recovered clock is used as the reference clock for the other ports. The way this typically works is that one port is set to gigabit slave. Its recovered clock is output onto a GPIO pin configured to SE_RCLK output. The SE_RCLK signal must be cleaned up via an external clock circuit. The cleaned up clock can then drive the SE_SCLK pin (Pin 128). The other ports are then configured as gigabit master and are configured to run using the SE_SCLK as the reference clock.

In order to prevent instability to the system, the port outputting its recovered clock on SE_RCLK must not use SE_SCLK as its reference clock.

There are two components to synchronous clocking support. The first is to output a recovered clock. The second is to select between the local reference clock and a cleaned up recovered clock.

9.9.1

Recovered Clock

The SE_RCLK0 and SE_RCLK1 signals can be configured to be output on any of the 15 available GPIO pins. See [Table 212 - Table 218](#) for configuration details. The SE_RCLK0 and SE_RCLK1 signals output a 25 MHz clock that is based on the 125 MHz recovered clock on the copper receive path when linked at 1000BASE-T or 100BASE-TX.

To configure a PHY port to output its recovered clock, Register 16_2.10 must be programmed to 0x1.

The source ports for the SE_RCLK0 and SE_RCLK1 signals are selected by configuring the TAI Global Configuration Register, offset 0x8 ([Table 269](#)). The SE_RCLK0 source port is selected by programming the PriRecClkSel bits [2:0]. The SE_RCLK1 source port is selected by programming the SecRecClkSel bits [6:4].

9.9.2

Clock Select



Note

The clock can only be recovered from a SERDES or PHY port. The clock cannot be recovered from an xMII port.

The 25 MHz reference clock source to the copper unit is independently selectable per port. On hardware reset XTAL_IN is selected as the reference clock source for all ports. SE_SCLK can be selected as the reference clock source on a per port basis.

For PHY ports, Register 16_2.7 selects whether the reference clock for the copper interface is based on XTAL_IN or SE_SCLK. Since changing the reference clocks disturbs the PHY, a software reset (Register 0_0.15) must be issued before any change to the clock select takes place.

For SERDES ports, register 26_1.15 selects whether the reference clock for the SERDES interface is based on XTAL_IN or SE_SCLK. Since changing the reference clock disturbs the SERDES, a software reset (Register 0_1.15) must be issued before any change to the clock select takes place.

- 0 = XTAL_IN,
- 1 = SE_SCLK.



Link Street®

88E6321/88E6320 Functional Specification

9.10

MDC/MDIO

The management interface provides access to the internal registers via the MDC and MDIO pins and is compliant with IEEE 802.3 Clause 22 and Clause 45 MDIO protocol. MDC is the management data clock input and, it can run from DC to a maximum rate of 12.5 MHz. At high MDIO fanouts the maximum rate may be decreased depending on the output loading. MDIO is the management data input/output and is a bi-directional signal that runs synchronously to MDC.

The MDIO pin requires a pull-up resistor in a range from 1.5 kohm to 10 kohm that pulls the MDIO high during the idle and turnaround.

PHY address is configured during the hardware reset sequence.

All the required serial management registers are implemented as well as several optional registers. A description of the registers can be found in the Register Description.

9.10.1

Clause 22 MDC/MDIO Management Interface

Typical read and write operations on the management interface are shown in [Figure 60](#) and [Figure 61](#).

Figure 60: Typical MDC/MDIO Read Operation

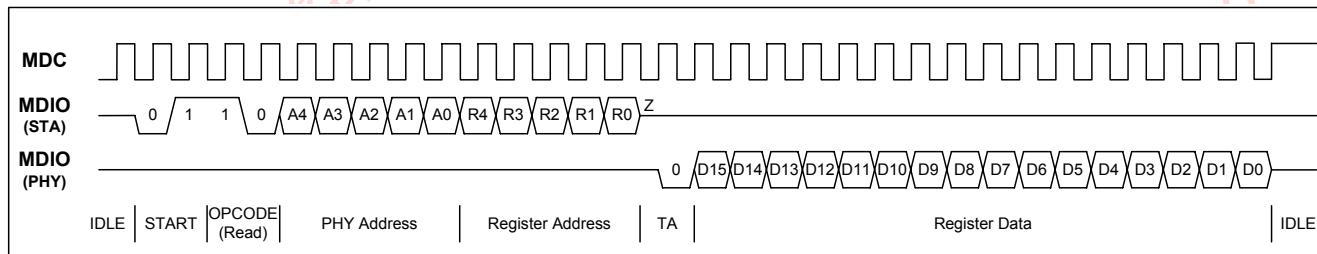
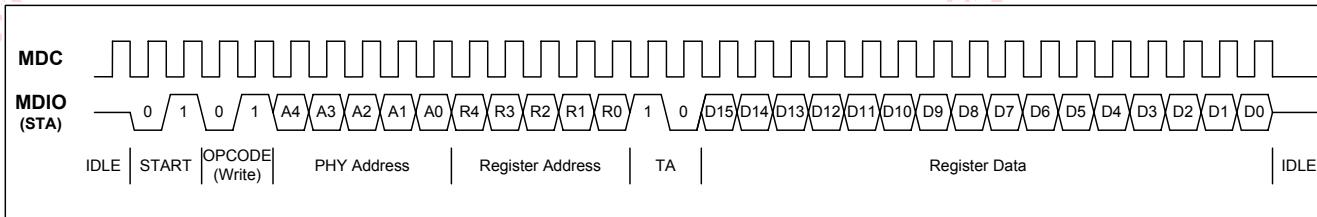


Figure 61: Typical MDC/MDIO Write Operation



[Table 55](#) is an example of a read operation.

Table 55: Serial Management Interface Protocol

32-Bit Preamble	Start of Frame	OpCode Read = 10 Write = 01	5-Bit PHY Device Address	5-Bit PHY Register Address (MSB)	2-Bit Turn around Read = z0 Write = 10	16-Bit Data Field	Idle
11111111	01	10	01100	00000	z0	0001001100000000	11111111

9.10.2 Extended Register Access

The IEEE defines only 32 registers address space for the PHY. In order to extend the number of registers address space available a paging mechanism is used. Register 22 bits 7 to 0 are used to specify the page. There is no paging for registers 22.

In this document, the short hand used to specify the registers take the form register_page.bit:bit, register_page.bit, register.bit:bit, or register.bit.

For example:

Register 16 page 2 bits 5 to 2 is specified as 16_2.5:2.

Register 16 page 2 bits 5 is specified as 16_2.5.

It takes four MDIO write commands to write the same register to the same value on all 4 ports. Register 22.15:14 can be used to selectively ignore PHYAD[4:2] and PHYAD[1:0] as shown in [Table 56](#) so that the same register address can be written to all four ports in one MDIO write command. PHYAD[4:0] will still be decoded for read commands.

Care must be taken to setup multiple port write. To enable the concurrent write access write register 22 four times in a row with bit 14 set to 1 – once to each PHYAD[4:0]. The values written on all 16 bits must be the same otherwise unpredictable behavior will occur.

Once the four write commands to register 22 are issued, all subsequent writes will be concurrent to all ports including writes to register 22.

Concurrent write access will continue as long as every write to register 22 sets 22.14 to 1.

To disable concurrent write access simply write register 22.14 to 0.

Table 56: Page Address

Register	Function	Setting	Mode	HW Rst	SW Rst
22.15:8	Reserved	00000000	RO	0	0
22.7:0	Page select for registers 0 to 21, 23 to 28	Page Number	R/W	00	Retain



Link Street®

88E6321/88E6320 Functional Specification

9.10.3

Clause 45 MDC/MDIO Management Interface (XMDIO)

Clause 45 provides extension of Clause 22 MDC/MDIO management interface to access more device registers while retaining its logical compatibility of the frame format. Clause 22 uses frame format with "Start of Frame" code of '01' while Clause 45 uses frame format with "Start of Frame" code of '00'. The extensions for Clause 45 MDIO indirect register accesses are specified in [Table 57](#).

Table 57: Extensions for Management Frame Format for Indirect Access

Frame	32-bit Preamble	Start of Frame	Opcode	5-bit PHY Address (MSB)	Device Address	2-bit Turnaround	16-bit ADDRESS/DATA Field	Idle
Address	1..1	00	00	PPPPP	DDDDD	10	AAAAAAAAAAAAAAA	Z
Write	1..1	00	01	PPPPP	DDDDD	10	DDDDDDDDDDDDDDDD	Z
Read	1..1	00	11	PPPPP	DDDDD	Z0	DDDDDDDDDDDDDDDD	Z
Read Increment	1..1	00	10	PPPPP	DDDDD	Z0	DDDDDDDDDDDDDDDD	Z

Clause 45 MDIO implements a 16-bit address register that stores the address of the register to be accessed. For an address cycle, it contains the address of the register to be accessed on the next cycle. For read, write, post-read increment-address cycles, the field contains the data for the register. At power up and reset, the contents of the register are undefined. Write, read, and post-read-increment-address frames access the address register, though write and read frames do not modify the contents of the address register.

9.10.4

Clause 22 Access to Clause 45 MDIO Manageable Device (MMD)

Clause 22 provides access to registers in a clause 45 MDIO MMD space using Register 13 and 14. Register 22:7:0 must be set to 0 to 7. If Register 22:7:0 is 8 to 255, the MMD registers are not accessible.

The MMD Access Control Register and Address/Data Register definitions are shown in [Table 58](#) and [Table 59](#).

Table 58: MMD Access Control Register

Page 0, Register 13

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Function	R/W	0x0	0x0	15:14 00 = Address 01 = Data, no post increment 10 = Data, post increment on reads and writes 11 = Data, post increment on writes only
13:5	Reserved	RO	0x000	0x000	Reserved
4:0	DEVAD	RO	0x00	0x00	Device address

Table 59: MMD Access Address/Data Register
Page 0, Register 14

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Address Data	R/W	0x0000	0x0000	If 13.15:14 = 00, MMD DEVAD's address register. Otherwise, MMD DEVAD's data register as indicated by the contents of its address register.

9.10.4.1 Write Operation

To write to the MMD register access:

1. To Register 13, write the Function field to 00 (address) and DEVAD field with the device address value;
2. To Register 14, write the MMD's register address value;
3. To Register 13, write the Function field to 01 (Data, no post increment) and DEVAD field to the same device address value (as step #1);
4. To Register 14, write the content to be written to the selected MMD's register

Step 1 and 2 can be skipped if the MMD's address register was previously configured.

9.10.4.2 Read Operation

To read from the MMD register access:

1. To Register 13, write the Function field to 00 (address) and DEVAD field with the device address value;
2. To Register 14, write the MMD's register address value;
3. To Register 13, write the Function field to 01 (Data, no post increment) and DEVAD field to the same device address value (as step #1);
4. From Register 14, read the content from the selected MMD's register.

Step 1 and 2 can be skipped if the MMD's address register was previously configured.

9.10.4.3 Write/Read Operation with Post Increment Function

Function '10' can be used to increment the address after each read and write access. Function '11' can be used to increment the address after write operation only. Function '11' enables a read-modify-write capability for successive addressed registers within the MMD.



9.10.5

Clause 45 Access to Clause 22 Registers

Clause 22 registers space can also be access through the Clause 45 MDIO protocol. All of the Clause 22 registers are mapped into Clause 45 Device Address (DEVAD) 3 vendor specific register space (0x8000 – 0x9FFF). The Clause 22 registers are mapped as the following:

$$\text{C45_REGAD}[15:0] = \{\text{3b}'100, \text{P22}[7:0], \text{C22_REGAD}[4:0]\}$$

Where:

P22[7:0] – Clause 22 register 22 paging

C22_REGAD[4:0] – Clause 22 REGAD[4:0]

C45_REGAD[15:0] – Clause 45 REGAD[15:0]

Table 60: Clause 45 Access to Clause 22 Registers Example

Clause 22 Registers		Clause 45 Registers	
Page	Register Address	Device Address	Register Address
0x0	0x4	0x3	0x8004 (3b'100, 8b'0000 0000, 5b'00100)
0x1	0x11 (Register 17)	0x3	0x8031 (3b'100, 8b'0000 0000, 5b'10001)
0x12	0x14 (Register 20)	0x3	0x8254 (3b'100, 8b'0001 0010, 5b'10100)

9.10.6

Preamble Suppression

The device is permanently programmed for preamble suppression. A minimum of one idle bit is required between operations.

10

Switch Register Description

The devices registers are accessible using the MDC_CPU and MDIO_CPU pins, which support the IEEE Serial Management Interface (SMI – Clause 22) used for PHY devices (Refer to MDC/MDIO in 88E6321/88E6320 Functional Specification Datasheet: Overview, Pinout, Applications, Mechanical and Electrical Specifications) or by using Ethernet frames using Remote Management ([Section 8](#)). The devices support two kinds of SMI address usage models. One uses 1 of the 32 possible Device addresses (when in Multi-chip mode - [Section 10.2, Multi-chip Addressing Mode](#)). The other uses all of the 32 possible Device addresses¹ (when in Single-chip mode - [Section 10.3, Single-chip Addressing Mode](#)). The device addresses and modes used are configurable at reset with the ADDR[4:0] configuration pins (ADDR[0] pins are unavailable externally and is considered zero - refer to "GMII/MII Transmit Interface," in 88E6321/88E6320 Functional Specification Datasheet: Overview, Pinout, Applications, Mechanical and Electrical Specifications).

Table 61: Register Map—Multi-Chip Addressing Mode

Address	Description	
00	SMI Command Register	page 212
01	SMI Data Register	page 212

Table 62: Register Map

Address	Description	
Switch Per-port Registers		
00	Port Status Register	page 216
01	Physical Control Register	page 221
02	Jamming Control Register	page 224
03	Switch Identifier Register	page 225
04	Port Control Register	page 226
05	Port Control 1	page 232
06	Port Based VLAN Map	page 232
07	Default Port VLAN ID & Priority	page 234
08	Port Control 2 Register	page 235
09	Egress Rate Control	page 239
10	Egress Rate Control 2	page 240
11	Port Association Vector	page 242
12	Port ATU Control	page 244
13	Priority Override Register	page 246
14	Policy Control Register	page 251

1. Any even address can be chosen.



Table 62: Register Map

Address	Description	
15	Port E Type	page 254
20–21	Reserved	
22	LED Control	page 255
	LED 0 & 1 Control, Register Index: 0x00 of LED Control for Ports 0 to 4	page 256
	Stretch and Blink Rate Control, Register Index: 0x06 of LED Control	page 259
	Port 3 Special Control, Register Index: 0x07 of LED Control on Port 3	page 260
	Port 4 Special Control, Register Index: 0x07 of LED Control on Port 4	page 260
	Port 5 Special Control, Register Index: 0x07 of LED Control on Port 5	page 261
23	Reserved	
24	Port IEEE Priority Remapping Registers	page 262
25	Port IEEE Priority Remapping Registers	page 262
27	Queue Counter Registers	page 263
28–29	Reserved	
30	Debug Counter	page 264
31	Cut Through Register	page 265
Switch Global 1 Registers		
00	Switch Global Status Register	page 267
01	ATU FID Register	page 268
02	VTU FID Register	page 269
03	VTU SID Register	page 269
04	Switch Global Control Register	page 270
05	VTU Operation Register	page 272
06	VTU VID Register	page 273
07	VTU/STU Data Register Ports 0 to 3 for VTU Operations	page 273
	VTU/STU Data Register Ports 0 to 3 for STU Operations	page 274
08	VTU/STU Data Register Ports 4 to 5 for VTU Operations	page 275
	VTU/STU Data Register Ports 4 to 5 for STU Operations	page 276
09	VTU/STU Data Register for VTU Operations	page 276
10	ATU Control Register	page 277
11	ATU Operation Register	page 278
12	ATU Data Register	page 280
13	ATU MAC Address Register Bytes 0 & 1	page 281
14	ATU MAC Address Register Bytes 2 & 3	page 281
15	ATU MAC Address Register Bytes 4 & 5	page 281
24	IEEE-PRI Register	page 282
25	IP Mapping Table	page 283
26	Monitor Control	page 284

Table 62: Register Map

Address	Description	
27	Total Free Counter	page 286
28	Global Control 2	page 287
29	Stats Operation Register	page 289
30	Stats Counter Register Bytes 3 & 2	page 292
31	Stats Counter Register Bytes 1 & 0	page 292
Switch Global 2 Registers		
00	Interrupt Source Register	page 294
01	Interrupt Mask Register	page 295
02	MGMT Enable Register 2x	page 296
03	MGMT Enable Register 0x	page 296
04	Flow Control Delay Register	page 297
05	Switch Management Register	page 298
06	Device Mapping Table Register	page 300
07	Trunk Mask Table Register	page 301
08	Trunk Mapping Table Register	page 301
09	Ingress Rate Command Register	page 302
10	Ingress Rate Data Register	page 303
11	Cross-chip Port VLAN Register	page 304
12	Cross-chip Port VLAN Data Register	page 305
13	Switch MAC/WoL/WoF Register	page 306
14	ATU Stats Register	page 314
15	Priority Override Table	page 315
16-19	Reserved	
20	EEPROM Command	page 319
21	EEPROM Data	page 319
22	AVB Command Register	page 320
23	AVB Data Register	page 321
24	SMI PHY Command Register	page 322
25	SMI PHY Data Register	page 322
26	Scratch and Misc. Register	page 323
	Scratch Byte 0, Register Index: 0x00 of Scratch and Misc. Control	page 323
	Scratch Byte 1, Register Index: 0x01 of Scratch and Misc. Control	page 323
	Voltage Regulator Control, Register Index: 0x06 of Scratch and Misc. Control	page 324
	EEE Timer Rates, Register Index: 0x0B of Scratch and Misc. Control	page 325
	EEE Wake Timer GE, Register Index: 0x0C of Scratch and Misc. Control	page 325
	EEE Assertion Timer, Register Index: 0x0D of Scratch and Misc. Control	page 325
	EEE WakeTimer, Register Index: 0x0E of Scratch and Misc. Control	page 326



Link Street®

88E6321/88E6320 Functional Specification

Table 62: Register Map

Address	Description	
26 (cont.)	EEE TxIdle Timer, Register Index: 0x0F of Scratch and Misc. Control	page 326
	GPIO 0 Port Stall Vector 0, Register Index: 0x20 of Scratch and Misc. Control - GPIO 14 Port Stall Vector 1, Register Index: 0x3D of Scratch and Misc. Control	page 326 - page 339
	GPIO Configuration, Register Index: 0x60 of Scratch and Misc. Control	page 340
	GPIO Configuration, Register Index: 0x61 of Scratch and Misc. Control	page 342
	GPIO Direction, Register Index: 0x62 of Scratch and Misc. Control	page 344
	GPIO Direction, Register Index: 0x63 of Scratch and Misc. Control	page 344
	GPIO Data, Register Index: 0x64 of Scratch and Misc. Control	page 345
	GPIO Data, Register Index: 0x65 of Scratch and Misc. Control	page 345
	GPIO Pin Control 0, Register Index: 0x68 of Scratch and Misc. Control - GPIO Pin Control 7, Register Index: 0x6F of Scratch and Misc. Control	page 346 - page 353
	CONFIG Data0, Register Index: 0x70 of Scratch and Misc. Control	page 354
	CONFIG Data1, Register Index: 0x71 of Scratch and Misc. Control	page 354
	CONFIG Data2, Register Index: 0x72 of Scratch and Misc. Control	page 355
	CONFIG Data3, Register Index: 0x73 of Scratch and Misc. Control	page 355
27	Watch Dog Control Register	page 356
28	QoS Weights Register	page 362
29	Misc Register	page 363
30	Misc Register	page 364
Port Ingress Rate Limiting (PIRL) Registers		
0	PIRL Bucket Configuration Register	page 366
1	PIRL Bucket Configuration Register	page 367
2	PIRL Bucket Configuration Register	page 367
3	PIRL Bucket Configuration Register	page 368
4	PIRL Bucket Configuration Register	page 368
5	PIRL Bucket Configuration Register	page 369
6	PIRL Bucket Configuration Register	page 369
7	PIRL Bucket Configuration Register	page 371
AVB Registers		
AVB - Precise Timing Protocol (PTP) Registers		
0	PTP Port Config Register	page 376
1	PTP Port Config Register	page 377
2	PTP Port Config Register	page 378
3-7	Reserved	
8	PTP Port Status Register	page 381
9	PTP Port Status Register	page 382
10	PTP Port Status Register	page 383
11	PTP Port Status Register	page 383

Table 62: Register Map

Address	Description	
12	PTP Port Status Register	page 384
13	PTP Port Status Register	page 386
14	PTP Port Status Register	page 386
15	PTP Port Status Register	page 387
16	PTP Port Status Register	page 387
17	PTP Port Status Register	page 389
18	PTP Port Status Register	page 389
19	PTP Port Status Register	page 390
21	PTP Port Status Register	page 390
AVB - PTP Global Registers		
0	PTP Global Config Register, AVBPort = 0xF	page 393
1	PTP Global Config Register, AVBPort = 0xF	page 393
2	PTP Global Config Register, AVBPort = 0xF	page 394
8	PTP Global Status Register, AVB = 0xF	page 394
AVB - PTP TAI Registers		
0	TAI Global Config Register, AVBPort = 0xE	page 396
1	TAI Global Config Register, AVBPort = 0xE	page 399
2	TAI Global Config Register, AVBPort = 0xE	page 400
3	TAI Global Config Register, AVBPort = 0xE	page 400
4	TAI Global Config Register, AVBPort = 0xE	page 401
5	TAI Global Config Register, AVBPort = 0xE	page 401
8	TAI Global Configuration Register, AVBPort = 0xE	page 402
9	TAI Global Config Register, AVBPort = 0xE	page 403
10	TAI Global Config Register, AVBPort = 0xE	page 404
11	TAI Global Config Register, AVBPort = 0xE	page 404
14	TAI Global Config Register, AVBPort = 0xE	page 405
15	TAI Global Config Register, AVBPort = 0xE	page 405
16	TAI Global Config Register, AVBPort = 0xE	page 405
17	TAI Global Config Register, AVBPort = 0xE	page 406
18	TAI Global Config Register, AVBPort = 0xE	page 406
19	TAI Global Config Register, AVBPort = 0xE	page 406
20	TAI Global Config Register, AVBPort = 0xE	page 407
21	TAI Global Config Register, AVBPort = 0xE	page 407
22	TAI Global Config Register, AVBPort = 0xE	page 407
23	TAI Global Config Register, AVBPort = 0xE	page 408
AVB - Policy		
0	AVB Policy Register	page 410



Link Street®

88E6321/88E6320 Functional Specification

Table 62: Register Map

Address	Description	
0	AVB Policy Global Clock Register, AVBPort = 0xF	page 414
4	AVB Policy Global Legacy Register, AVBPort = 0xF	page 415
8	AVB Policy Global Limit Register, AVBPort = 0xF	page 415
12	AVB Policy Global OUI Register Byte 0 & 1, AVBPort = 0xF	page 416
13	AVB Policy Global OUI Register Byte 2, AVBPort = 0xF	page 416
AVB - Qav Port Config		
0	QavPort Config Register	page 418
1	QavPort Config Register	page 418
2	QavPort Config Register	page 419
3	QavPort Config Register	page 419
4	QavPort Config Register	page 419
5	QavPort Config Register	page 420
6	QavPort Config Register	page 420
7	QavPort Config Register	page 420
AVB - Qav Global		
0	Qav Global Config Register, AVBPort = 0xF	page 422
8	Qav Global Status Register, AVBPort = 0xF	page 423
9	Qav Global Status Register, AVBPort = 0xF	page 423
12	Qav Global Status Register, AVBPort = 0xF	page 424
13	Qav Global Status Register, AVBPort = 0xF	page 424
Switch Global 3 Registers - TCAM		
0	TCAM Operation Register	page 426
TCAM Page 0		
2	Keys Register 1	page 429
3	Keys Register 2	page 430
4	Keys Register 3	page 431
5	Keys Register 4	page 432
6	Match Data Register 1	page 433
7	Match Data Register 2	page 433
TCAM Page 1		
0	TCAM Operation Register	page 435
2	Match Data Register 23	page 435
3	Match Data Register 24	page 435
TCAM Page 2		
0	TCAM Operation Register	page 437
2	Action Register 1	page 437
3	Action Register 2	page 438

Table 62: Register Map

Address	Description	
4	Action Register 3	page 439
5	Action Register 4	page 440
28	TCAM Debug Register	page 442
31	TCAM Debug Register	page 442



Link Street®

88E6321/88E6320 Functional Specification

10.1 Register Types

The registers in the devices are made up of one or more fields. The way in which each of these fields operate is defined by the field's Type. The function of each Type is described below.

Type	Description
LH	Register field with latching high function. If status is high, then the register bit is set to one and remains set until a read operation is performed through the management interface or a reset occurs.
LL	Register field with latching low function. If status is low, then the register bit is cleared to zero and remains zero until a read operation is performed through the management interface or a reset occurs.
RES	Reserved for future use. All reserved bits are read as zero unless otherwise noted.
RO	Read only.
R/W	Read and write with initial value indicated.
RWR	Read/Write reset. All field bits are readable and writable. After reset, register field is cleared to zero.
RWS	Read/Write set. All field bits are readable and writable. After reset, register field is set to a non-zero value specified in the text.
SC	Self-Clear. Writing a one to this register causes the desired function to be immediately executed, then the register field is automatically cleared to zero when the function is complete.
Update	Value written to the register field does not take effect until soft reset is executed; however the written value can be read even before the software reset.
Retain	Value written to the register does not take effect without a software reset and the computer maintains its value after a software reset.
WBAR	Write back as read. All fields must be read and left unchanged before performing a write.
WO	Write only. Reads to this type of register field return undefined data.

10.2

Multi-chip Addressing Mode

When Multi-chip addressing mode is used on the SMI interface, the devices respond to only 1 of the 32 possible SMI device addresses so that it can share the SMI interface with multiple devices. The SMI address that is used is determined by the ADDR[4:0] configuration pins. As there is no way to set ADDR[0], any even SMI address can be selected. In this mode, only two of the devices' registers are directly accessible, the SMI Command register ([Table 63](#)) and the SMI Data register ([Table 64](#)). These two registers are used to access all the other device registers indirectly (along with any PHY registers that may be attached to it).

Indirect accessing of the other devices registers is accomplished by setting the SMI Command register's DevAddr and RegAddr bits to point to the devices register to access. Use the DevAddr and RegAddr values defined for devices in the Single-chip Addressing mode ([Section 10.3](#)).

Multi-chip Addressing Mode is enabled when the ADDR[4:0] configuration pins (See [88E6321/88E6320 Functional Specification Datasheet: Overview, Pinout, Applications, Mechanical and Electrical Specifications](#)) carry a non-zero value at the rising edge of RESETn. The ADDR[4:0] configuration pins also define the single SMI address to which this device will respond to. To avoid conflicts, this requires that all devices on the same SMI interface use unique ADDR[4:0] values. An SMI address of 0x00 is not supported in this mode as this value on the ADDR[4:0] pins places the devices into Single-chip Addressing mode ([Section 10.3](#)).



Link Street®

88E6321/88E6320 Functional Specification

Table 63: SMI Command Register¹

Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	SMIBusy	SC	Internal SMI Unit Busy. This bit must be set to a one to start an internal SMI operation (see SMIOp below). Only one SMI operation can be executing at one time so this bit must be zero before setting it to a one. When the requested SMI operation is completed, this bit is automatically be cleared to a zero.
14:13	Reserved	RES	Reserved for Future Use
12	SMIMode	RWR	Internal SMI Mode bit. This bit is used to define the SMI frame type to generate as follows: 0 = Generate IEEE 802.3 Clause 45 SMI frames ² 1 = Generate IEEE 802.3 Clause 22 SMI frames ³
11:10	SMIOp	RWR	Internal SMI Opcode. These bits are used to select the SMI opcode to operate on during SMI commands as follows: When the SMIMode bit = 1 then SMIOp = (IEEE 802.3 Clause 22): 11 = Reserved 10 = Read Data Register 01 = Write Data Register 00 = Reserved When the SMIMode bit = 0 then SMIOp = (IEEE 802.3 Clause 45): 11 = Read Data Register with post increment on the address register 10 = Read Data Register 01 = Write Data Register 00 = Write Address Register
9:5	DevAddr	RWR	Internal SMI Device Address bits. These bits are used to select the SMI device (Clause 22) or port (Clause 45) to operate on during SMI commands.
4:0	RegAddr	RWR	Internal SMI Register Address bits. These bits are used to select the SMI register (Clause 22) or device class (Clause 45) to operate on during SMI commands.

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

2. Clause 45 SMI frames can be used to access Clause 45 devices connected to the MDC_PHY and MDIO_PHY pins

3. Clause 22 SMI frames must be used to access the internal switch registers

Table 64: SMI Data Register¹

Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15:0	SMIData	RWR	SMI Data register. During SMI writes these bits must be written with the SMI data to be written prior to starting the SMI operation (i.e., before setting SMIBusy to a one). During SMI reads these bits will contain the SMI data that was read after the SMI read operation is completed (i.e., SMIBusy returns to a zero). Writes to this register must not be done while SMIBusy is a one.

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

10.3

Single-chip Addressing Mode

Figure 62 shows the register map in Single-chip mode assuming the single chip mode is being used. In this mode, the devices respond to all 32 SMI device addresses so it must be the only device connected to a SMI Master (typically a CPU). The devices use SMI device addresses internally and all unused SMI device addresses are ignored.

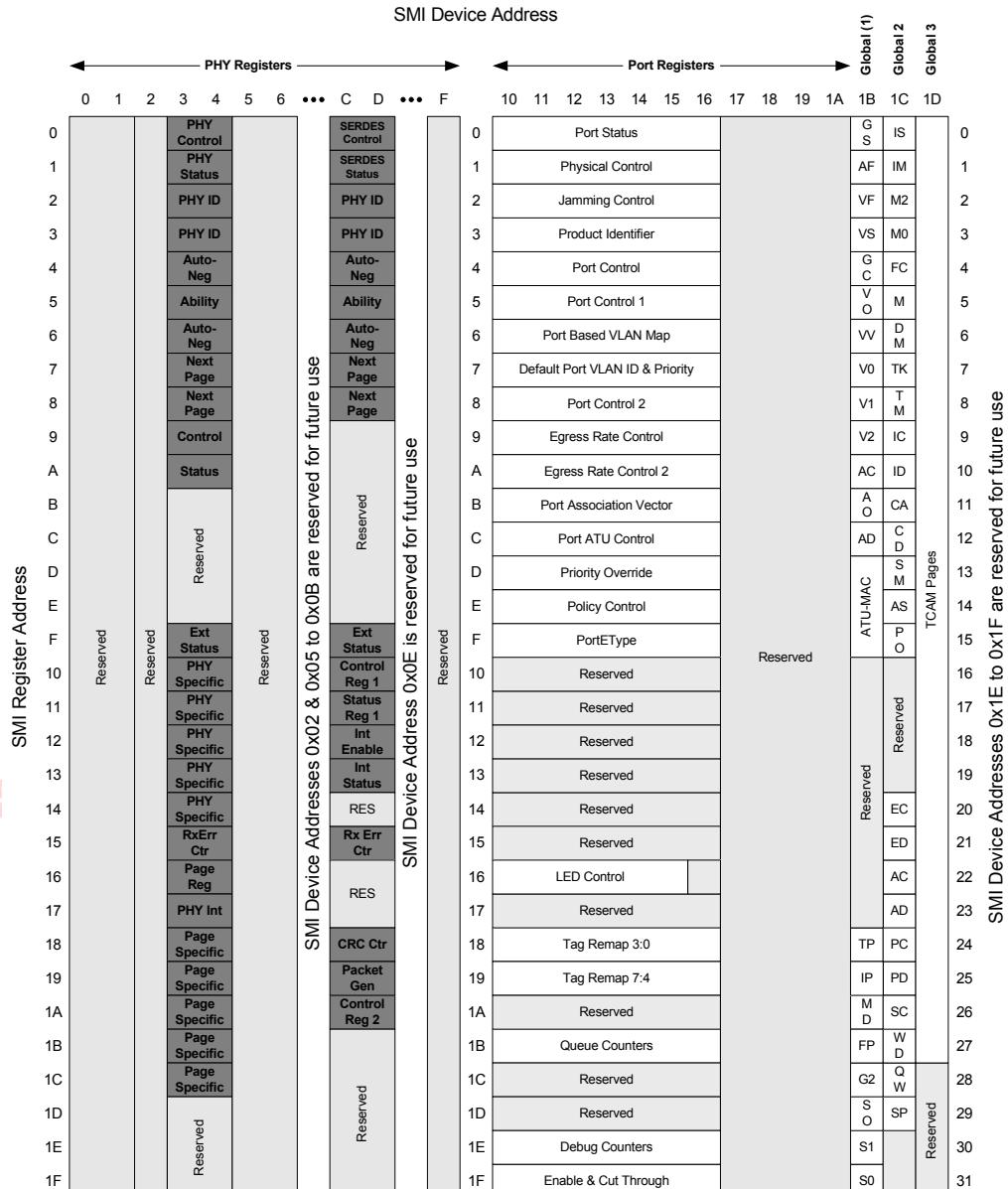
If external PHYs are attached to Ports 0, 1, 2, 5 and/or Port 6, the ports must use SMI device addresses 0x0 (for Port 0), 0x1 (for Port 1), 0x2 (for Port 2), 0x5 (for Port 5) to 0x6 (for Port 6) so the PPU can automatically poll the PHYs for PHYs for link, speed, duplex and flow control status, and communicate the current PHY state to the correct MAC. The registers in the external PHYs can be accessed by using this device's SMI PHY Command and Data registers (Global 2, offsets 0x18 & 0x19).

Single-chip Addressing Mode is enabled when the ADDR[4:0]n configuration pins (See 88E6321/88E6320 Functional Specification Datasheet: Overview, Pinout, Applications, Mechanical and Electrical Specifications) are all zeroes at the rising edge of RESETn.

**Note**

The internal PHY and SERDES registers in Figure 62 are shown in grey because they are accessed at the above SMI device address indirectly using the SMI PHY registers (Global 2 offset 0x18 and 0x19). The internal Port 3's & Port 4's PHYs are mapped at SMI device addresses 0x03 & 0x04 respectively. The internal SERDES on Ports 0 and Port 1 are mapped at SMI device addresses 0x0C and 0x0D respectively.

Figure 62: Device Register Map



GS = Global Status
AF = ATU FID
VF = VTU FID
VS = VTU SID
GC = Global Control

IS = Interrupt Source
IM = Interrupt Mask
M2 = MGMT Enables 2x
M0 = MGMT Enables 0x
FC = Flow Control Delays
M = Management

VO = VTU Operation
VV = VTU VID
V0 = VTU Data Ports 3:0
V1 = VTU Data Ports 7:4
V2 = VTU Data Ports A:8

DM = Device Mapping
TK = Trunk Mask
TM = Trunk Mapping
IC = Ingress Rate Command
ID = Ingress Rate Data
CA = Cross Chip Port VLAN Addr

AC = ATU Control
AO = ATU Operation
AD = ATU Data
TP = IEEE Tag Priority
IP = IP PRI Mapping Table
MD = Monitor Destinations

CD = Cross Chip Port VLAN Data
SM = Switch MAC
AS = ATU Stats
PO = Priority Overrides
ID = EEPROM Cmd & Data
EC & ED = EEPROM Cmd & Data
AC & AD = AVB Cmd & Data

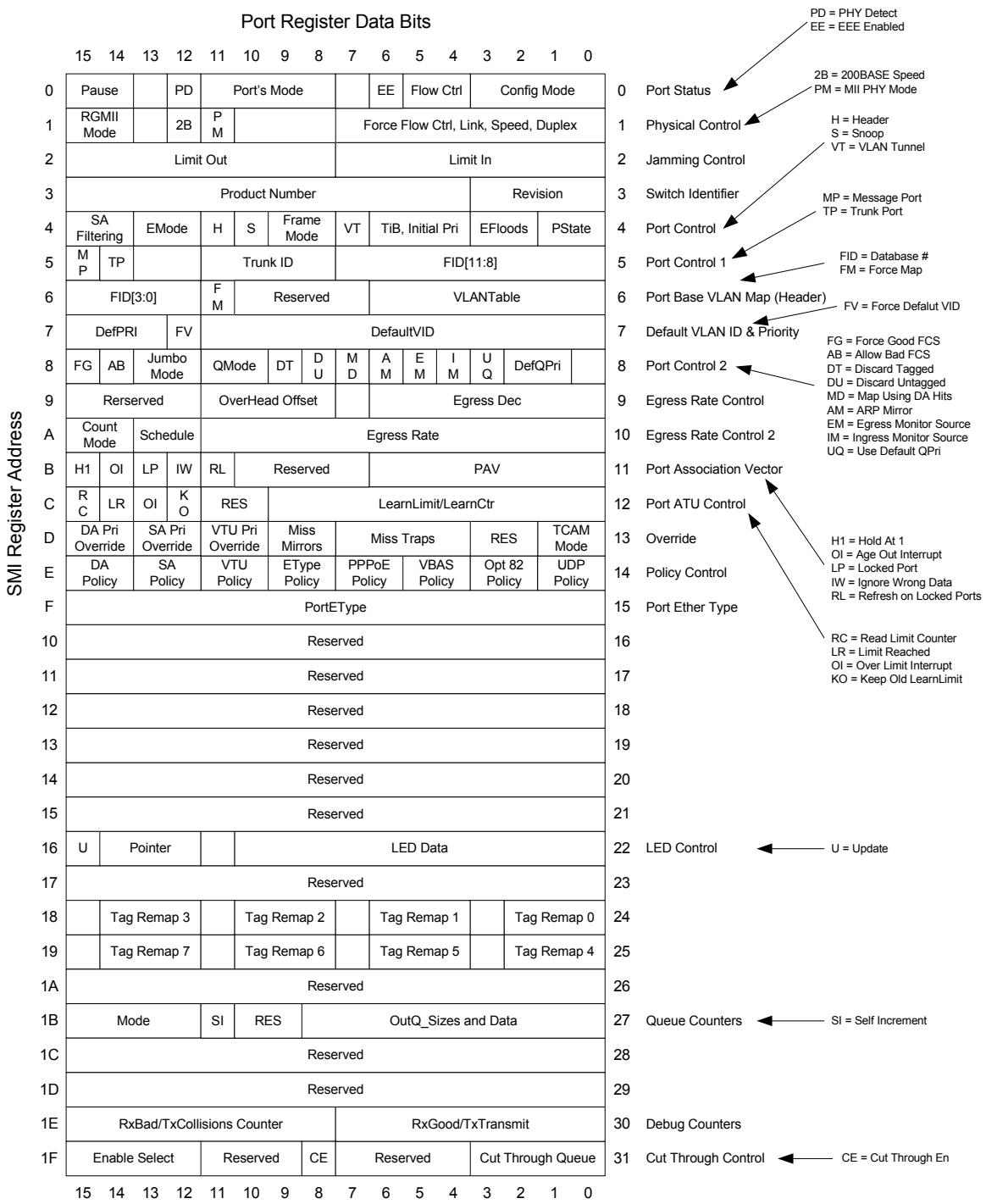
FP = Free Pool
G2 = Global Control 2
SO = Stats Operation
S1 = Stats Data Bytes 3:2
S0 = Stats Data Bytes 1:0

PC = SMI PHY Command
PD = SMI PHY Data
SC = Scratch
WD = Watch Dog Control
QW = QoS Weights
SP = SDET Polarity

10.4 Switch Port Registers

Each Ethernet port in the devices contain their own per port registers. Each per port register is 16-bits wide and their bit assignments are shown in Figure 63.

Figure 63: Per Port Register Bit Map





Link Street®

88E6321/88E6320 Functional Specification

Table 65: Port Status Register¹
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	PauseEn	RO	<p>Pause Enabled bit. This indicates that full-duplex flow control will be used on this port if the port is in full-duplex mode. It is valid when the Link bit is a one. This bit reflects the Pause result from auto-negotiation only (i.e., the ForceFlowControl bit's value is not reflected in this bit).</p> <p>0 = MAC Pause not implemented in the link partner or in MyPause 1 = MAC Pause is implemented in the link partner and in MyPause</p>
14	MyPause	RO	<p>My Pause bit. This bit is sent to the PHY during PHY Polling Unit (PPU) initialization. This bit is not meaningful on ports that do not support Auto-Negotiation (i.e., internal ports). It is set high if FLOW (See 88E6321/88E6320 Functional Specification Datasheet: Overview, Pinout, Applications, Mechanical and Electrical Specifications) is high during RESETn.</p> <p>0 = MAC Pause is not to be advertised as supported in this port and half-duplex back pressure will not be used on this port. 1 = MAC Pause is to be advertised as supported in this port and half-duplex back pressure will be used on this port if this port is in half-duplex mode</p> <p>This bit reflects the Reset value of the FLOW pin. Its value is not changed by forcing flow control on or off on this port.</p>
13	Reserved	RES	Reserved for future use.
12	PHYDetect	RWR	<p>802.3 PHY Detected. This bits is set to a one if the PPU finds a non-all-one's value in either SMI registers 2 or 3 at the SMI device address 0x10 less than this address.</p> <p>0 = An 802.3 PHY is not attached to this port 1 = An 802.3 PHY is attached to this port</p> <p>If the SERDES select configuration pin (S_SEL) is 0x0 after Reset, Auto Media selection between copper and fiber is enabled on Port 4 as long as this register bit is a one. Clearing this bit to a zero in this case will turn off Auto Media and allow Port 4 to link to the SERDES only (see Table 4 on page 16).</p> <p>These bits are set at the end of the PPU's init routine (that is why this register must not be written to while the PPUState is Initializing – i.e., 01). The PPU poll routine uses these bits to determine which port's to poll the PHYs for Link, Duplex, Speed and Flow Control. If software changes these bits, the PPU changes its polling accordingly. A SWReset (Table 99) restarts the PPU's initialization and causes a re-write to this register (also in Table 99).</p>

Table 65: Port Status Register¹
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
11	Link	RO	<p>Link Status. This bit indicates the link status of the port as follows:</p> <p>0 = Link is down 1 = Link is up</p> <p>The port's Link bit mirrors the LinkValue bit if the link is being forced by ForcedLink being a one (Table 67). Otherwise the port's Link bit takes the value of the source defined in Table 66.</p> <p>For Port 2, Port 5, and Port 6 (the GMII/MII/RGMII interfaces), the port's Px_ENABLE represents the link status. The Px_ENABLEs come from any available GPIO pin – see Enable Select (Port offset 0x1F).</p>
10	Duplex	RO	<p>Duplex mode. This bit is valid when the Link bit, above, is set to a one.</p> <p>0 = Half-duplex 1 = Full-duplex</p> <p>The port's Duplex bit takes the value of the DpxValue bit if the duplex is being forced by ForcedDpx being a one (Table 67). Otherwise the port's Duplex bit takes the value of the source defined in Table 66.</p>
9:8	Speed	RO	<p>Speed mode. These bits are valid when the Link bit, above, is set to a one. When the port is configured in (G)MII mode the Speed bits are determined by the Px_MODE[2:0] bits.</p> <p>00 = 10 Mbps 01 = 100 or 200 Mbps 10 = 1000 Mbps 11 = Reserved for future use</p> <p>The port's Speed bits take the value of ForceSpeed bits if the speed is being forced by ForceSpeed being non-0x3 (Table 67). Otherwise the port's Speed bits take the value of source defined in Table 66.</p> <p>NOTE: These bits will reflect the actual speed of Ports 0, 1, 2, Port 5 and 6 only when an external PHY is attached to the port. Otherwise these bits will reflect the C_Mode speed of the port (as best it can) if the speed is not being forced.</p>
7	Reserved	RES	Reserved for future use.
6	EEE Enabled	RO	<p>EEE (Energy Efficient Ethernet) Enabled from the PHY. This bit is set to one when the PHY on this port is advertising support for EEE and its link partner also advertises support for EEE. This bit only valid on ports with integrated PHYs. This bit will always be zero on the MII ports.</p> <p>This bit is the ANDing of the two signals from the PHY (local and link partner EEE advertise bits). This bit is not affected if EEE is forced on or off (Port offset 0x01).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 65: Port Status Register¹

Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
5	TxPaused	RO	Transmitter Paused. This bit is set to a one whenever the Rx MAC receives a Pause frame with a non-zero Pause time that is used by the Tx MAC (i.e., the transmitter is paused off). If the port is in half-duplex mode, this bit is never a one since all Rx Pause frames are ignored. If the port is in full-duplex mode this bit is never a one if Rx Pause frames are ignored because flow control is disabled on this port.
4	FlowCtrl	RO	Flow Control. This bit is set to a one whenever the Rx MAC determines that no more data should be entering this port. If the port is in half-duplex mode, this bit's being a one indicates that the port is using back pressure. If the port is in full-duplex mode this bit's being a one indicates that the port is going to or has sent a Pause frame with a non-zero Pause time to its link partner.
3:0	C_Mode (not all modes available on all ports)	RO or RW	<p>Config Mode. These bits return the switch port's current interface type configuration mode. Most of these modes are determined at reset based on the physical design of the device and/or the port's Px_MODE configuration (CONFIG) bit values at reset. Basically, a port's C_Mode bits equal the port's Px_MODE (for ports that have Px_MODE configuration pins) after reset.</p> <p>The only ports whose C_Mode bits may change after reset is Port 2 and 6. If P6_MODE = 0x6 (disabled) during reset, then both port 2 and port 6 come up in the xmII Tristate (disabled) mode. In this case, the C_Mode bits for ports 2 and 6 can be written to so that the two interfaces can be configured independently of each other. Configure Port 6 first followed by Port 2. If Port 6 is configured in C_Mode 0x1, 0x2 or 0x3 then Port 2 will stay at C_Mode 0x6. All other combinations are valid.</p> <p>Port 2's C_Mode is set to the value on the P6_MODE configuration pins after reset if P6_MODE <> 0x1, 0x2 or 0x3². If P6_MODE = 0x1, 0x2 or 0x3 then Port 2's C_Mode is set to 0x6 (disabled).</p> <p>Port 0's C_Mode is set to 0x8 if P0_SMODE is low during reset. Port 1's C_Mode is set to 0x8 if P1_SMODE is low during reset. Else their C_Modes will be 0x9 or 0xA depending upon the port's PHY Detect bit (bit 12 of this register).</p>

- 1.
2. The C_Mode bits are normally read only. But if P6_MODE is configured to be 0x6 (disabled) during reset, then the C_Mode bits for Port 2 and 6 can be written to. This allows the two interfaces to be configured independently if needed. See [Table 66](#) to understand what each C_Mode value means.

The source of the port's Mode, Link, Speed and Duplex bits for each possible C_Mode (assuming no forcing of the bits is occurring— [Table 67](#)) is defined in [Table 66](#). Each of these bits, Link, Speed and Duplex, can be individually forced to any value by using the Forcebits in [Table 67](#).

Not all C_Mode values are supported by all the ports. The list of ports that support each C_Mode is noted under the Switch Port's column in [Table 66](#).

Table 66: Interface Configuration Matrix

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

1. The digital xMII's are Tristated if their Link goes down by the Px_ENABLE pin (see Port offset 0x1F).
2. The digital xMII's are Tristated if their Link goes down by the Px_ENABLE pin (see Port offset 0x1F).
3. C_Modes 0x0, 0x1, 0x4 and 0x5 on Ports 2, 5 and 6 (valid combinations only) default to a Speed of 100 but they can be forced to 10 Mbit operation (see Port offset 0x01). For C_Mode 0x1 and C_Mode 0x0 when in PHY mode, the Clock Mode's frequency will also change accordingly..
4. All digital xMII's mode defaults to full duplex operation, but 10 or 100 speed modes can be forced into half duplex operation (see Port offset 0x01) with the exception of C_Mode 0x0 which can support full duplex only.
5. C_Mode 0x0 starts out as MAC mode where INCLK and OUTCLK are inputs. But the interface can be reconfigured to PHY mode where INCLK and OUTCLK are outputs (see Port offset 0x01).
6. The Link Partner's Input clocks determine the actual speed of the MII. The Speed bits (see Port offset 0x00) will not reflect the actual speed of the port unless software updates the ForceSpd bits (Port offset 0x01).
7. The 100BASE-FX mode defaults to full-duplex operation, but 10 or 100 speed modes can be forced into half-duplex operation (see Port offset 0x01).



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED



Note

C_Mode is initially set to the port's Px_MODE (see Px_OUTD pin description). C_Mode can be modified by software on Ports 2 and 6 if P6_MODE is 0x6.



Note

All invalid Px_MODE or C_Mode setting result in a Disabled port (e.g., setting Port 5's Px_MODE to 0x3).

Table 67: Physical Control Register
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15	RGMII Rx Timing (valid on Port 2, Port 5 and Port 6 only)	RWR	<p>RGMII Receive Timing Control. Changes to this bit are disruptive to normal operation. Hence any change to this register must be done only while the port's link is down (see bits 5:4 below).</p> <p>0 = Default 1 = Add delay to RXCLK for IND inputs when port is in RGMII mode</p> <p>See Part 1 of 3: Overview, Pinout, Applications, Mechanical, and Electrical Specifications, Section 3.8.1, RGMII Timing for Different RGMII Modes for RGMII Timing Modes.</p>
14	RGMII Tx Timing (valid on Port 2, Port 5 and Port 6 only)	RWR	<p>RGMII Transmit Timing Control. Changes to this bit are disruptive to normal operation. Hence any change to this register must be done only while the port's link is down (see bits 5:4 below).</p> <p>0 = Default 1 = Add delay to GTXCLK for OUTD outputs when port is in RGMII mode</p> <p>See Part 1 of 3: Overview, Pinout, Applications, Mechanical, and Electrical Specifications, Section 3.8.1, RGMII Timing for Different RGMII Modes for RGMII Timing Modes.</p>
13	Reserved	RES	Reserved for future use.
12	200BASE (valid on Port 2, Port 5, and Port 6 only)	RWR	<p>200 BASE Mode. When Port2, 5 or 6's C_Mode is 0x0 or 0x1 (Port offset 0x00) this bit can be used along with the ForceSpd bits below to force the port's speed as follows:</p> <p>0 = 100 Base speed (25 MHz MII CLK) 1 = 200 Base speed (50 MHz MII CLK)</p>
11	MII PHY (valid on Port 2, Port 5, and Port 6 only)	RWR	<p>MII PHY Mode. When port 2, 5 or 6's C_Mode is 0x0 (Port offset 0x00) this bit can be used to configure the port to MAC or PHY mode as follows:</p> <p>0 = MII MAC mode (Px_INCLK and Px_OUTCLK are inputs) 1 = MII PHY mode (Px_INCLK and Px_OUTCLK are outputs)</p>
10:8	Reserved	RES	Reserved for future use.
7	FCValue	RWR	<p>Flow Control's Forced value. This bit is used to force flow control (if full-duplex) or backpressure (if half-duplex) to be enabled when the ForcedFC bit (below) is set to a one. Flow control/back pressure is forced enabled when this bit is set to a one. It is forced disabled when this bit is cleared to a zero. If the ForcedFC bit (below) is cleared to a zero, this bit has no effect.</p> <p>If Ingress Pause limiting is enabled (port offset 0x03) then this bit will be cleared to a zero if the Pause limit was reached on this port – so do not write to this register while Ingress Pause Limiting is enabled on this port – unless the write is intended to re-enable Pausing on this Jammed port.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 67: Physical Control Register

Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
6	ForcedFC		<p>Force Flow Control. When this bit is set to a one flow control (if full-duplex) or backpressure (if half-duplex) for this port is forced to the value in the FCValue register (above) regardless of what the normal flow control value would be. In this way, flow control/backpressure can be forced to be enabled or disabled. When this bit is cleared to a zero, normal flow control detection occurs.</p> <p>If Ingress Pause limiting is enabled (port offset 0x03) then this bit will be set to a one if the Pause limit was reached on the port – so do not write to this register while Ingress Pause Limiting is enabled on this port – unless the write is intended to re-enable Pausing on this Jammed port.</p>
5	LinkValue	RWR	Link's Forced value. This bit is used to force the link up or down when the ForcedLink bit (below) is set to a one. The link will be forced up when this bit is set to a one. It will be forced down when this bit is cleared to a zero. If the ForcedLink bit (below) is cleared to a zero this bit has no effect.
4	ForcedLink	RWR	<p>Force Link. When this bit is set to a one the link for this port's MAC is forced to the value in the LinkValue register (above) regardless of what the normal link's value would be. In this way, the link can be forced to be up or down. When this bit is cleared to a zero, normal link detection occurs.</p> <p>This bit forces the MAC to consider Link to be up or down. It does not force Link up or down in the PHY, but it will on the MII pins. If software needs to stop the flow of frames on a port it is better to use the Disabled port state in the Port State bits in the Port Control registers (offset 0x04).</p>
3	DpxValue	RWR	Duplex's Forced value. This bit is used to force the link to full- or half-duplex mode, when the ForcedDpx bit (below) is set to a one. The link duplex is forced to full when this bit is set to a one. It will be forced to half when this bit is cleared to a zero (Do not try to force half-duplex mode in 200BASE and faster modes- those modes are not supported and results are unpredictable). If the ForcedDpx bit (below) is cleared to a zero this bit has no effect.
2	ForcedDpx	RWR	Force Duplex. When this bit is set to a one the duplex for this port's MAC will be forced to the value in the DpxValue register (above) regardless of what the normal duplex's value would be. In this way the duplex can be forced to be full or half. When this bit is cleared to a zero, normal duplex detection occurs.

NOTE: Only change the port's duplex when its link is down.

Table 67: Physical Control Register
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
1:0	ForceSpd	RWS to 0x3	<p>Force Speed. These bits are used to force the speed on this port's MAC as follows:</p> <p>00 = 10 Mbps 01 = 100 or 200 Mbps (depending upon the 200BASE bit above) 10 = 1000 Mbps 11 = Speed is not forced. Normal speed detection occurs.</p> <p>NOTE: Only change the port's speed when its link is down. These bits change the speed of the port's MAC interface only (and its pins on ports where the port's MAC interface is exposed as a xMII interface¹). If this port is connected to a PHY and the PHY is not at the same speed as the MAC unpredictable results will occur.</p>

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



The duplex and speed on a port must not be changed unless the link on the port is down.



Link Street®

88E6321/88E6320 Functional Specification

Table 68: Jamming Control Register
Offset: 0x02 or Decimal 2

Bits	Filed	Type	Register
15:8	LimitOut	RWS To 0xFF	<p>Limit the number of continuous Pause refresh frames transmitted that can be transmitted from this port – assuming each Pause refresh is for the maximum pause time of 65536 slot times. When full-duplex Flow Control is enabled on this port, these bits are used to limit the number of Pause refresh frames that can be generated from this port to keep this port's link partner from sending any data.</p> <p>Clearing these bits to 0x00 will allow continuous Pause frame refreshes to egress this port as long as this port remains congested.</p> <p>Setting these bits to 0x01 will allow 1 Pause frame to egress from this port for each congestion situation.</p> <p>Setting these bits to 0x02 will allow up to 2 Pause frames to egress from this port for each congestion situation, etc.</p> <p>The upper 3-bits of this register are a 2n multiplier for the lower 5 bits. The maximum count is $2^7 * 31$ (7 comes from the upper 3 bits while 31 comes from the lower 5 bits). This equals $128 * 31$ or 3,968 maximum Pause times.</p>
7:0	LimitIn	RWR	<p>Limit the number of continuous Pause refresh frames that can be received on this port (if full-duplex) or the number of 16 consecutive collisions (if half-duplex). When a port has flow control enabled, these bits can be used to limit how long this port can be Paused or Back Pressured off to prevent a port stall through jamming.</p> <p>When these bits are in the range of 0x01 to 0xFF, and a frame is ready to be transmitted out this port, but it can't be transmitted due to the port being jammed, this limit mechanism starts. The limit mechanism starts counting new Pause refresh frames (Pause frames that re-load the Pause timer to other than 0x0000) or counts of 16 consecutive collisions. If the counter reaches the value contained in this register, the Port's ForceFC bit will be set to a one and its FCValue bit will be cleared to a zero and the global JamLimit Interrupt (Global 2, offset 0x00) will be set. This effectively disables Flow Control on the port once the Pause timer expires. If a frame gets transmitted out this port before the counter reaches this limit (i.e., the frame that was ready to be transmitted that started this process gets transmitted) then this limit mechanism counter resets back to zero.</p> <p>If the port is in half-duplex mode and Flow Control is disabled on the port, the JamLimit Interrupt will still be generated if the limit is reached on a frame. If Discard Excessive is set to a one (Global 1, offset 0x04) then the JamLimit Interrupt will never occur on half-duplex ports (since the frame is discarded after 16 consecutive collisions).</p> <p>Setting these bits to 0x00 will allow continuous jamming to be received on this port without the Port's ForceFC and FCValue bits getting modified.</p> <p>The modification of this Port's ForceFC and FCValue bits is the only indication that the limit was reached on this port.</p>

Table 69: Switch Identifier Register
Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:4	Product Num	RO	Product Number or identifier: 88E6321 = 0x320 88E6320 = 0x125
3:0	Rev	RO	Revision Identifier. The initial version of the devices has a Rev of 0x0. This Rev field may change at any time. Contact Marvell® FAEs for current information on the device revision identifier.



Link Street®

88E6321/88E6320 Functional Specification

Table 70: Port Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15:14	SA Filtering	RWR	<p>Source Address Filtering controls. These bits select the SA (Source Address) filtering method to be used on the port as follows:</p> <p>00 = SA Filtering Disabled – no frame will be filtered (i.e., discarded) due to the contents of its Source Address field</p> <p>01 = Drop On Lock. Ingressing frames will be discarded if their SA field is not in the ATU's address database (i.e., its a new or unknown Source Address) or if this port's bit is not set in the PortVec bits for the frame's SA (i.e., this port is not the source port for that MAC address). Used for MAC based 802.1X.</p> <p>10 = Drop On Unlock. Ingressing frames will be discarded if their SA field is in the ATU's address database as a Static entry with a PortVec of all zeros. Used to discard frames from known untrusted sources.</p> <p>11 = Drop to CPU. Ingressing frames will be mapped to the CPUDest (global offset 0x1A) if their SA field is in the ATU's address database as a Static entry with a PortVec of all zeros and the frame is not otherwise filtered. Otherwise, the frames will be discarded if their SA field is not in the ATU's address database (i.e., its a new or unknown Source Address) or if this port's bit is not set in the PortVec bits for the frame's SA (i.e., this port is not the source port for that MAC address). This mode is a form of MAC based 802.1X where some frames can be forced to the CPU for further authentication prior to full authorization.</p> <p>SAFiltering[0] needs to zero when hardware address learn limiting is enabled on the port (Port ATU Control, offset 0x0C). In other words, hardware address learn limiting will work with either SA Filtering Disabled, or with SA Filtering set to Drop on Unlock.</p>

Table 70: Port Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
13:12	Egress Mode	RWR	<p>Egress Mode. These bits determine how frames look when they egress this port.</p> <p>The effect of these bits is controlled by the Frame Mode bits below (bits 9:8 of this register) as follows:</p> <p>When Frame Mode = (00) Normal Network Frames these bits define the default tagging mode of the egressing frames. The default mode is used when the VID assigned to the frame during ingress is not contained in the VTU. The default modes are:</p> <ul style="list-style-type: none"> 00 = Default to Unmodified mode – frames are transmitted unmodified¹ 01 = Default to Transmit all frames Untagged – remove the tag from any tagged frame 10 = Default to Transmit all frames Tagged – add a tag to any untagged frame 11 = Reserved for future use. <p>When Frame Mode = (01) DSA Tag Frames these bits must remain at 00 as all other modes are ‘Reserved for future use’.</p> <p>When Frame Mode = (10) Provider Tag Frames these bits must remain at 00 as all other modes are ‘Reserved for future use’.</p> <p>When Frame Mode = (11) Ether Type DSA Tag Frames these bits define which frames get Ether Type DSA Tagged. In this case all Control frames egress with an Ether Type DSA Tag regardless of the setting of these bits (Control frames are To_CPU, From_CPU and To_Sniffer). Non-Control frames (i.e., Forward DSA Tag frames) will egress Ether Type DSA Tagged if these bits are 0b11, otherwise Forward frames will egress as Normal Network Frames as follows:</p> <ul style="list-style-type: none"> 00 = Egress Forward DSA frames as Unmodified Normal Network Frames 01 = Egress Forward DSA frames as Untagged Normal Network Frames 10 = Egress Forward DSA frames as Tagged Normal Network Frames 11 = Egress all frames from this port with an Ether Type DSA Tag
11	Header	RWR	<p>Ingress & Egress Header Mode. When this bit is set to a one all frames egressing this port are pre-pended with the Marvell® 2-byte Egress Header just before the frame’s DA field. Also, all frames ingressing this port are expected to be pre pended with the Mavell 2-byte Ingress Header just before the frame’s DA field. On Ingress the 1st 2 bytes after the SFD (Start of Frame Delimiter) are removed from the frame and the frame’s CRC and size is recomputed. If the frame’s Ingress Header is non-zero it is used to update the port’s VLAN Map register value (Port offset 0x06). When this bit is cleared to a zero, normal Ethernet frames egress the switch and are expected to ingress the switch.</p> <p>Header mode is intended to be used only on a port that is directly connected to a CPU that is performing routing as the Layer 3 portion of the frame becomes 32-bit aligned in the CPU’s memory. It can be used in conjunction with DSA or Ether Type DSA Frame Modes (see Frame Mode bits below).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 70: Port Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
10	IGMP/MLD Snoop	RWR	<p>IGMP and MLD Snooping. When this bit is set to a one and this port receives an IPv4 IGMP frame or an IPv6 MLD frame, the frame is switched to the CPU port² overriding the destination ports determined by the DA mapping³. When this bit is cleared to a zero IGMP/MLD frames are not treated specially.</p> <p>IGMP/MLD Snooping is intended to be used on Normal Network or Provider ports only (see Frame Mode bits below) and only if Cut Through is disabled on the port (Port offset 0x1F) as the IPv6 Snoop point may be after byte 64.</p>
9:8	Frame Mode	RWR	<p>Frame Mode. These bits are used to define the expected Ingress and the generated Egress tagging frame format for this port as follows:</p> <p>00 = Normal Network 01 = DSA (Distributed Switch Architecture) 10 = Provider 11 = Ether Type DSA</p> <p>00 = Normal Network mode uses industry standard IEEE 802.3ac Tagged or Untagged frames. Tagged frames use an Ether Type of 0x8100. Ports that are expected to be connected to standard Ethernet devices should use this mode.</p> <p>01 = DSA mode uses a Marvell® defined tagged frame format for Chip-to-Chip and Chip-to-CPU connections. The extra data placed in the frame is needed to support the Spanning Tree Protocol (STP) as well as cross-chip features like Trunks, Mirrors, etc. Ports that are interconnected together to form a larger switch and ports connected to the management CPU must use this mode.</p> <p>10 = Provider mode uses user definable Ether Types per port (see PortEType register, port offset 0x0F) to define that a frame is Provider Tagged. Ports that are connected to standard Provider network devices, or devices that use Tagged frames with an Ether Type other than 0x8100 should use this mode.</p> <p>Frames that ingress this port with an Ether Type that matches the port's PortEType register will be considered tagged (for the DiscardTag and Discarded Untagged discarding policy - Port offset 0x08), will have the tag's VID and PRI bits assigned to the frame (i.e., they will be used for switching and mapping), and will have the Provider Tag removed from the frame. If subsequent Provider Tags are found following the 1st Provider Tag, they too will be removed from the frame with their VID and PRI bits being ignored (if Remove1Tag is 0 in the Management register, Global (2), offset 0x05). Modified frames will be padded if required.</p> <p>Frames that ingress this port with an Ether Type that does not match the port's PortEType will be considered untagged (for the DiscardTag and Discarded Untagged discarding policy - Port offset 0x08). The ingressing frames are modified so they are ready to egress out Customer ports (Normal Network Frame Mode ports) unmodified.</p>

Table 70: Port Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
9:8 (cont.)	Frame Mode (cont.)	RWR	<p>Frames that egress this port will always have a tag added (even if they were already tagged). The added tag will contain this port's PortEType as its Ether Type. The PRI bits will be the Frame Priority (FPri) assigned to the frame during ingress. The VID bits will be the source port's Default VID bits (if the source port was in Normal Network mode), or the VID assigned to the frame during ingress (if the source port was in Provider mode or if the frame was DSA Tagged).</p> <p>11 = Ether Type DSA mode uses standard Marvell® DSA Tagged frame information following a user definable Ether Type. This mode allows the mixture of Normal Network frames with DSA Tagged frames and is useful on ports that connect to a CPU.</p> <p>Frames that ingress this port with an Ether Type that matches the port's PortEType will be considered DSA Tagged and processed accordingly. The frame's Ether Type and DSA pad bytes will be removed so the resulting frame will be ready to egress out Marvell DSA Tag Mode ports unmodified. Frames that ingress this port with a different Ether Type will be considered Normal Network Frames and processed accordingly.</p> <p>Marvell DSA Tag control frames (To_CPU, From_CPU and To_Sniffer) that egress this port will always get the port's PortEType inserted followed by two pad bytes of 0x00 before the DSA Tag. Marvell DSA Tag Forward frames that egress this port can egress just like the control frames (with the added Ether Type and pad) or they can egress as if the port was configured in Normal Network mode. This selection is controlled by the port's EgressMode bits above.</p>
7	VLAN Tunnel	RWR	<p>VLAN Tunnel. When this bit is cleared to a zero the port based VLANs defined in the VLANTable (Table 72 - Port offset 0x06), 802.1Q VLANs defined in the VTU (if 802.1Q is enabled - Table 70 and Table 100 QMode in Port offset 0x08 and the VTU in Global 1 offsets 0x02 and 0x09) and Trunk Masking (Table 128 - Global 2 offset 0x07) are enforced for ALL frames. When this bit is set to a one the port based VLANTable masking, 802.1Q VLAN membership masking and the Trunk Masking is bypassed for any frame entering this port with a DA that is currently 'static' in the ATU. This applies to unicast as well as multicast frames.</p> <p>NOTE: Do not set this bit to a 0x1 if the port's AVBTunnel bit (AVB Port offset 0x00) is set to a 0x1.</p>
6	TagIfBoth	RWS	<p>Use Tag information for the initial QPri assignment if the frame is both tagged and its also IPv4 or IPv6 and if InitialPri, below, = 0x3, Use Tag & IP Priority.</p> <p>The initial QPri is assigned as follows: 0 = QPri is frame's DiffServ bits (for IPv4) or the frame's Traffic Class bits (for IPv6) mapped using the IP PRI Mapping registers (Global 1 offsets 0x10 to 0x17). 1 = QPri is the determined FPri mapped using the IEEE PRI Mapping register (Global 1, offset 0x18).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 70: Port Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
5:4	InitialPri	RWS to 0x3	<p>Initial Priority assignment. Each frame entering a port is assigned a Frame Priority (FPri) and a Queue Priority (QPri). The FPri determined during ingress is written to the frame's IEEE 802.3ac tag PRI bits if the frame egresses tagged or if the frame egresses Provider Tagged (see EgressMode bits in – port offset 0x04). The QPri is used internally to determine which egress priority queue the frame is mapped into.</p> <p>On DSA ports (see FrameMode bits - port offset 0x04) the frame's default FPri is the DSA tag's PRI bits and the default QPri is FPri[2:1]. On non-DSA ports this register is used to select the frame's initial FPri & QPri depending upon the frame's type and content.</p> <p>These initial FPri & QPri assignments can be overridden by various frame type overrides (Global 2, offset 0x0F) and/or content overrides if enabled on the port (see port offset 0x0D, but FPri content Overrides should not be enabled on DSA ports). If a frame does not meet the condition listed in the following table the defaults are assigned to frame.</p> <p>On non-DSA ports the default FPri is the port's DefFPri bit in the Default VLAN ID and Priority register at Port offset 0x07. The default QPri is obtained by mapping the frame's FPri value using the IEEE PRI Mapping register (Global 1, offset 0x18).</p> <p>The initial FPri and QPri on non-DSA ports are assigned as follows:</p> <ul style="list-style-type: none">00 = Use Port defaults for FPri and QPri.01 = Use Tag Priority. If the frame is tagged, FPri is set to the frame's tag PRI bits re-mapped by the port's Tag Remap registers (Port offsets 0x17 & 0x18) and the QPri is the determined FPri mapped by the IEEE PRI Mapping register (Global 1, offset 0x18). If the frame is untagged the port defaults are used for FPri and QPri.10 = Use IP Priority. If the frame is IPv4 or IPv6, QPri is the frame's DiffServ bits (for IPv4) or the frame's Traffic Class bits (for IPv6) mapped by the IP PRI Mapping registers (Global 1, offsets 0x10 to 0x17) and FPri[2:1] is the frame's QPri and FPri[0] is the port's DefFPri[0] unless UselpFPri is set to a one (Global 1 offset 0x19), in which case FPri is set by mapping the frame's DiffServ or Traffic Class bits into the IP Mapping Table (Global 1 offset 0x019). If the frame is not IPv4 nor IPv6 the port defaults are used for FPri and QPri.11 = Use Tag & IP Priority. If the frame is tagged, FPri is the frame's tag PRI bits re-mapped by the port's Tag Remap registers (Port offsets 0x17 & 0x18). If the frame is also IPv4 or IPv6 QPri's value will be determined by the TagIfBoth bit above. If the frame is untagged but it is IPv4 or IPv6, FPri and QPri are set according to the Use IP Priority setting above. If the frame is neither tagged nor IPv4 nor IPv6 the port defaults are used for FPri and QPri.

Table 70: Port Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
3:2	Egress Floods	RWS to 0x3	<p>Egress Flooding mode. The DA of every frame, unicast and multicast, is searched in the ATU. If the DA is found in the address database it is considered known. If it is not found it is considered unknown. Frames with known DA's are not effected by this register.</p> <p>Frames with unknown DA's generally flood out all the ports (except for the port it originally came in on). This register can be used to prevent frames with unknown DA's from egressing this port as follows:</p> <ul style="list-style-type: none"> 00 = Do not egress any frame with an unknown DA (unicast or multicast) 01 = Do not egress any frame with an unknown multicast DA 10 = Do not egress any frame with an unknown unicast DA 11 = Egress all frames with an unknown DA (unicast and multicast) <p>The FloodBC (flood broadcast) bit in the Global 2 Management register (global 2, offset 0x05) is used to determine if Broadcast frames are considered multicast frames in the above description.</p>
1:0	PortState	RWR Or RWS to 0b11 ⁴	<p>Port State. These bits are used to manage a port to determine what kind of frames, if any, are allowed to enter or leave a port for simple bridge loop detection or 802.1D Spanning Tree or other 802.1 protocols. The state of these bits can be changed at any time without disrupting frames currently in transit.</p> <p>The Port States are:</p> <ul style="list-style-type: none"> 00 = Disabled. The switch port is disabled and it will not receive or transmit any frames. The QC returns any pre-allocated ingress queue buffers when the port is in this mode. 01 = Blocking/Listening. The switch will examine all frames without learning any SA addresses, and will discard all but MGMT frames. It will allow MGMT frames only to exit the port. This mode is used for BPDU handling for bridge loop detection and Spanning Tree support or other 802.1 protocols. 10 = Learning. The switch will examine all frames, learning all SA address (except those from MGMT⁵ frames), and still discard all but MGMT frames. It will allow MGMT frames only to exit the port. 11 = Forwarding. The switch examines all frames, learning SAs from all good frames (except those from MGMT frames), and receives and transmits all frames as a normal switch.

1. If this port has 802.1Q disabled and Cross-chip Port Based VLANs are being used in the switch, this port's EgressMode must be Default to Normal mode (to ensure the frames egress the switch looking exactly how they entered the switch) or Always add a Tag (to ensure the frames egress the switch with an extra tag compared to how they entered the switch).
2. The CPU port is determined by the CPUDest bits in Monitor Control Register (Global Offset 0x1A).
3. IGMP/MLD frames that are ingress filtered will not be sent to the CPUDest port.
4. The PortState bits for all ports come up in the Forwarding state unless the NO_CPU configuration pin is a zero after reset, the CPU attached mode, in which case the ports come up Disabled and all internal PHYs and SERDES come up powered down. Ports 0 and 1 always come up in the Disabled port state in the small package regardless of the state of the NO_CPU configuration pin (software can subsequently change their Port State value).
5. MGMT (management) frames are the only kind of frames that can be tunneled through Blocked ports. A MGMT frame is any frame whose multicast DA address appears in the ATU Database with the MGMT Entry State, or whose DA is an enabled special multicast (see Rsvd2CPU in Global 2 offset 0x05).



Link Street®

88E6321/88E6320 Functional Specification

Table 71: Port Control 1
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	Message Port	RWR	Message Port. When the Learn2All bit in the ATU is set to a one (Table 107 - Global 1 offset 0x0A) learning message frames will be generated. These frames will be sent out all ports whose Message Port is set to a one. If this feature is used it is recommended that all DSA Tag ports, except for the CPU's port, have their MessagePort bit set to a one. Ports that are not DSA Tag ports (i.e., normal Network ports) should not have their MessagePort bit set to a one.
14	Trunk Port	RWR	Trunk Port. When this bit is set to a one, this port is considered to be a member of a Trunk with the Trunk ID defined below. When this bit is set to a zero, the port is treated as an individual port.
13:12	Reserved	RES	Reserved for future use.
11:8	Trunk ID	RWR	Trunk ID. When the Trunk Port bit (above) is set to a one these bits define which trunk this port is to be associated with. All ports that are members of the same trunk must be assigned the same Trunk ID and each group of ports that form a trunk must be assigned unique Trunk IDs.
7:0	FID [11:4]	RWR	Port's Default Filtering Information Database (FID) bits 11:4. This field can be used with non-overlapping VLANs to keep each VLAN's MAC address mapping database separate from the other VLANs. This allows the same MAC address to appear multiple times in the address database (at most one time per VLAN) with a different port mapping per entry. This field is overridden by the FID returned from a VTU hit and it should be zero if not used. It must be a unique number for each independent, non-overlapping, address database if used. The lower four bits of the port's default FID are contained in the Port Base VLAN Map register (Table 72 - Port offset 0x06).

Table 72: Port Based VLAN Map¹
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description
15:12	FID[3:0]	RWR	Port's Default Filtering Information Database (FID) bits 3:0. This field can be used with non-overlapping VLANs to keep each VLAN's MAC address mapping database separate from the other VLANs. This allows the same MAC address to appear multiple times in the address database (at most one time per VLAN) with a different port mapping per entry. This field is overridden by the FID returned from a VTU hit and it should be zero if not being used. It needs to be a unique number for each independent, non-overlapping, address database, if used. The upper 4 bits of the port's default FID are contained in the Port Control 1 register (Table 71 - Port offset 0x05).

Table 72: Port Based VLAN Map¹ (Continued)
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description
11	ForceMap	RWR	<p>Force Mapping. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one all received frames will be considered MGMT² and they are mapped to the port or ports defined in the VLANTable bits below overriding the mapping from the address database. The forcing function is needed to get BPDU frames to egress specific ports by the CPU for the Spanning Tree Protocol. ForceMapped frames will egress ports that are not in the Disabled port state (i.e., they are MGMT frames and will egress out Blocked ports). This bit is accessible by the CPU's Ingress Header so the CPU can enable and disable MGMT and forcing on a frame by frame basis.</p> <p>NOTE: Learning is disabled on MGMT frames, so this bit being set to a one, also disables learning on frames entering this port.</p>
10:7	Reserved	RES	Reserved for future use.
6:0	VLANTable	RWS to all ones except for this port's bit	<p>In Chip Port based VLAN Table. The bits in this table are use to restrict which output ports this input port can send frames to. The VLANTable bits are used for all frames, except for MGMT frames³, even if 802.1Q is enabled on this port (Port offset 0x08). These bits restrict where a port can send frames to (unless a VLANTunnel frame is being received – Table 70 - Port offset 0x04, or an AVBTunnel frame is being received - AVB Port offset 0x00). If ForceMap is set to a one, these bits indicate which port or ports all frames that ingress this port are sent to overriding the mapping from the address database.</p> <p>To send frames to Port 0, bit 0 of this register must be a one. To send frames to Port 1, bit 1 of this register must be a one, etc. After reset, all ports are accessible since all the other port number bits are set to a one. This Port's bit is zero after reset. This prevents frames leaving the port on which they arrived. This Port's bit can to be set to a one in the devices, which allows frames to be switched back to the port on which they arrived. In view of this fact, care should be taken in writing code to manipulate these bits.</p> <p>This register is reset to 0x7FE for Port 0 (SMI Device Address 0x10), and it resets to 0x7FD for Port 1 (Addr 0x11), to 0x7FB for Port 2 (Addr 0x12), etc.</p> <p>Cross Chip Port base VLANs are supported by the Cross Chip Port VLAN Table (Global 2 offsets 0x0B and 0x0C).</p>

- The contents of this register can be modified on a frame by frame basis if the port's Header Mode is enabled (Port offset 0x04).
 NOTE: Only the lower four bits of the FID field can be modified by the Header. Software that controls the FID field by using the Marvell® Header needs to take this into account. The DefaultVIDs used for Cross-Chip Port Based VLANs must be unique from the VIDs used for the 802.1Q VLANs currently active in the switch. Port Based VLAN ports need to have their frame's egress unmodified or the internal VID will be added to the frame if it is set to egress tagged.
- MGMT = Management frames, frames that can tunnel through Blocked ports (see PortStates in Port offset 0x04).
- See [Section 6.3.1](#).



Link Street®

88E6321/88E6320 Functional Specification

Table 73: Default Port VLAN ID & Priority
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:13	DefPri	RWR	Default Frame Priority. The bits of this register are used as the default ingress priority to use when no other priority information is available (neither is the frame IEEE Tagged, nor is it an IPv4 nor an IPv6 frame—or the frame is a priority type that is currently disabled (see InitialPri, port offset 0x04). These are the initial Frame priority (FPRI) bits assigned to the frame. The Ingress priority determines the Egress Queue the frame is switched into (the initial QPRI) by the DefPri bits being re-mapped by the IEEE-PRI Register (Port offset 0x18— Table 113).
12	Force DefaultVID	RWR	Force to use Default VID. When 802.1Q is enabled on this port (Port offset 0x08) and this bit is set to a one, all Ingress frame's VID are ignored and the DefaultVID below is assigned and replaced into the frame (if the frame egresses tagged – Egress Mode, Port offset 0x04). When this bit is cleared to a zero all IEEE 802.3ac Tagged frames with a non-zero VID use the frame's VID unmodified. When 802.1Q is disabled on this port, this bit has no effect.
11:0	DefaultVID	RWS to 0x001	Default VLAN Identifier. When 802.1Q is enabled on this port the DefaultVID field is used as the IEEE Tagged VID added to untagged or priority tagged frames during egress that ingressed from this port. It is also used as a tagged frame's VID if the frame's VID was 0x000 (i.e., it is a priority tagged frame) or if the port's Force DefaultVID bit (above) is set to a one. When 802.1Q is disabled on this port, the DefaultVID field is assigned to all frames entering the port (if they are tagged or untagged ¹). This assignment is used internal to the switch, so only that Cross-chip Provider ports can be supported.

1. Port Based VLAN ports (ports where 802.1Q is disabled, port offset 0x08) need to have their frame's egress unmodified (Egress Mode, Port offset 0x04) or the internal VID will be added to the frame if it is set to egress tagged.

Table 74: Port Control 2 Register
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15	ForceGoodFCS	RWR	Force good FCS in the frame. When this bit is cleared to a zero frames entering this port must have a good CRC or else they are discarded. When this bit is set to a one the last four bytes of frames received on this port are overwritten with a good CRC and the frames are accepted by the switch (assuming that the frame's length is good and it has a destination).
14	AllowBadFCS	RWR	Allow receiving frames on this port with a bad FCS. When this bit is cleared to a zero the normal action of discarding ingress frames with CRC errors occurs on this port.
13:12	Jumbo Mode	RWS to 0x2 ¹	<p>JumboMode bits determine the maximum frame size (aka MTU) allowed to be received or transmitted from or to a given physical port.</p> <p>0x0 - Receive and Transmit frames with max byte count of 1522. 0x1 - Receive and Transmit frames with max byte count of 2048 0x2 - Receive and Transmit frames with max byte count of 10240 0x3 - Reserved</p> <p>This implies that a Jumbo frame may be allowed to be received from a given input port but may or may not be allowed to be transmitted out of a port or ports.</p> <p>For example if Port 2's JumboMode is 0x1 and it receives a 2100 Bytes frame, then the ingress pipe discards the frame as an oversized frame and InOversize MIB counter is updated.</p> <p>Another example is if a Jumbo frame, say 4500 Bytes long, is destined to go to ports 3 and 4 and Port 3's JumboMode is 0x2 and Port 4's JumboMode is 0x0, then the frame gets sent to Port 3 but not to Port 4.</p> <p>NOTE: The definition of frame size is counting the frame bytes from MAC-DA through Layer2 CRC of the frame.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 74: Port Control 2 Register

Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
11:10	802.1QMode	RWR	<p>IEEE 802.1Q Mode for this port. These bits determine if 802.1Q based VLANs are used along with port based VLANs (Port offset 0x06) for this Ingress port. It also determines the action to be taken if an 802.1Q VLAN Violation is detected. VLAN barriers (both port based and 802.1Q based) can be bypassed by VLANTunnel (Port offset 0x04), MGMT (Section 6.1) and AVB_Tunnel - AVB Policy Port offset 0x0) frames.</p> <p>These bits work as follows:</p> <p>00 = 802.1Q Disabled. Use Port Based VLANs only. The VLANTable bits (Port offset 0x06) and the Cross-chip Port VLAN Table (Global 2, offsets 0x0B & 0x0C) determine which Egress ports this Ingress port is allowed to switch frames to for all frames² (i.e., the frame's VID is ignored for switching and it's VID is not altered in the frame, i.e., all frames are considered untagged even if they are IEEE tagged). The VID assigned to the frame is the port's DefaultVID (port offset 0x07) which is used as the VID in the Provider Tag if the frame egresses a Provider port (see Frame Mode, port offset 0x04).</p> <p>01 = Fallback. Enable 802.1Q for this Ingress port. Do not discard Ingress Membership violations³ and use the VLANTable bits (i.e., port based VLANs Port offset 0x06) below if the frame's VID is not contained in the VTU (both errors are logged – Table 100 - Global 1 offset 0x05).</p> <p>10 = Check. Enable 802.1Q for this Ingress port. Do not discard Ingress Membership violation but discard the frame if its VID is not contained in the VTU (both errors are logged – Table 100 - Global 1 offset 0x05).</p> <p>11 = Secure. Enable 802.1Q for this Ingress port. Discard Ingress Membership violations and discard frames whose VID is not contained in the VTU (both errors are logged – Table 100 - Global 1 offset 0x05).</p>
9	Discard Tagged	RWR	<p>Discard Tagged Frames. When this bit is set to a one all non-MGMT frames that are processed as tagged are discarded as they enter this switch port. Priority only tagged frames (with a VID of 0x000) are considered untagged. This feature works if 802.1Q is enabled on the port or not (802.1Q Mode bits above).</p> <p>If the port is configured in Provider Mode (Frame Mode in Port Control, port offset 0x04) and this bit is set to a one, frames that contain an Ether Type that matches the port's PortEType (port offset 0x0F) that have a non-zero VID will be discarded.</p> <p>Discard Tagged should not be set on DSA or Ether Type DSA ports (see Frame Mode, port offset 0x04).</p>

Table 74: Port Control 2 Register
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
8	Discard Untagged	RWR	<p>Discard Untagged Frames. When this bit is set to a one all non-MGMT frames that are processed as untagged are discarded as they enter this switch port. Priority only tagged frames (with a VID of 0x000) are considered untagged. This feature works if 802.1Q is enabled on the port or not (802.1Q Mode bits above).</p> <p>If the port is configured in Provider Mode (Frame Mode in Port Control, port offset 0x04) and this bit is set to a one, frames that don't contain an Ether Type that matches the port's PortEType (port offset 0x0F) and frames that contain an Ether Type that matches the port's PortEType that have a zero VID will be discarded.</p> <p>Discard UnTagged should not be set on DSA or Ether Type DSA ports (see Frame Mode, port offset 0x04).</p>
7	MapDA	RWS	<p>Map using DA hits. When this bit is set to a one, normal switch operation occurs where a frame's DA is used to direct the frame out of the correct ports. When this bit is cleared to a zero the frame will be sent out of the ports defined by EgressFloods (port offset 0x04) even if the DA is found in the address database.</p> <p>If a multicast or unicast frame's DA is contained in the ATU with a MGMT Entry State the frame will be mapped out the port(s) defined by the ATU entry (i.e., the setting of the MapDA bit is ignored for MGMT frames).</p>
6	ARP Mirror	RWR	<p>ARP Mirror enable. When this bit is set to a one non-filtered Tagged or Untagged Frames that ingress this port that have the Broadcast Destination Address with an Ethertype of 0x0806 are mirrored to the CPUDest port (Global 1 offset 0x1A). This mirroring takes place after the ingress mapping decisions to allow ARPs to get to a CPU that is otherwise isolated. When this bit is cleared to a zero no special ARP handling will occur.</p> <p>ARP Mirror should not be set on DSA or Ether Type DSA ports (see Frame Mode, port offset 0x04) or extra mirroring will result.</p>
5	Egress Monitor Source	RWR	<p>Egress Monitor Source Port. When this bit is cleared to a zero, normal network switching occurs. When this bit is set to a one any frame that egresses out this port will also sent to the EgressMonitorDest Port (Table 115 - Global 1 offset 0x1A).</p> <p>The 802.1Q mode and VTU entries on the Egress Monitor Destination Port must be set the same as they are on the Egress Monitor Source port so the frames egress with the same tagged or untagged information.</p> <p>Egress Monitor Source should not be set on DSA or Ether Type DSA ports unless the port is directly connected to a CPU port and the CPU's code is being debugged (see Frame Mode, port offset 0x04).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 74: Port Control 2 Register

Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
4	Ingress Monitor Source	RWR	<p>Ingress Monitor Source Port. When this bit is cleared to a zero normal network switching occurs. When this bit is set to a one, any frame that ingresses this port is also sent to the IngressMonitorDest Port (Table 115 - Global 1 offset 0x1A). The frame is sent to the IngressMonitorDest Port even if it is discarded owing to switching policy (like VLAN membership, etc.) but the frame will not be forwarded if its contains an error (such as CRC, etc.) or is filtered by ingress rate limiting (Global 2 offsets 0x09 and 0xA).</p> <p>Ingress Monitor Source should not be set on DSA or Ether Type DSA ports unless the port is directly connected to a CPU port and the CPU's code is being debugged (see Frame Mode, port offset 0x04).</p>
3	Use Def Qpri	RWR	Use Default Queue Priority. When this bit is cleared to a zero the initial Qpri (Queue Priority) assigned to a frame is determined by the port's InitialPri and TagIfBoth settings (Port offset 0x04). When this bit is set to a one the initial Qpri assigned to the frame is the port's DefQPri (bits 2:1 below).
2:1	DefQPri	RWR	Default Queue Priority. The value of these bits are used as the Port's default queue priority and may be assigned to frames entering this port based on the UseDefQPri bit (above).
0	Reserved	RES	Reserved for future use.

1. The JUMBO configuration pin provides the power on reset value for these bits.
2. The VLANTable is sufficient to define Port Based VLANs when only one device is being used in a system (i.e., the VLANTable works for in-chip port based VLANs). When multiple devices are used in a system the Cross-Chip Port VLAN table (global 2, offset 0x0B & 0x0C) is used for frames entering a DSA port (or Ether Type DSA port if the Forward frames are DSA tagged, see Frame Mode, port offset 0x04). Both of these tables can be used with 802.1Q enabled using the VTU for frame VID switching (i.e., both port based and Q based VLAN are supported at the same time, in-chip and cross-chip).
3. VTU or 802.1Q Ingress Membership violations occur when the source port is not a member of the frame's VID

Table 75: Egress Rate Control
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:8	Frame Overhead	RWR	<p>Egress Rate Frame Overhead adjustment.</p> <p>This field is used to adjust the number of bytes that need to be added to a frame's IPG on a per frame basis. This is to compensate for a protocol mismatch between the sending and the receiving stations. For example if the receiving station were to add more encapsulations to the frame for the nodes further down stream, this per frame adjustment would help reduce the congestion in the receiving station.</p> <p>The egress rate limiter multiplies the value programmed in this field by four for computing the frame byte offset adjustment value (i.e., the amount the IPG is increased for every frame). This adjustment, if enabled, is added to the Egress Rate Control's calculated transmitted byte count (Port offset 0x0A) meaning Egress Rate Control must be enabled for this Frame Overhead adjustment to work¹.</p> <p>The egress overhead adjustment can add the following number of byte times to each frame's transmitted byte count: 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56 and 60.</p> <p>Example: If FrameOverhead = 0xB the egress rate limiter would increase the IPG/transmitted byte count² between every frame by an additional 44 bytes.</p> <p>When the Count Mode (port offset 0x0A) is in Frame based egress rate shaping mode, these Frame Overhead bits must be 0x0.</p>
7	Reserved	RES	Reserved for future use.
6:0	Egress Dec	RWS 0x01	<p>Egress Rate Decrement value.</p> <p>These bits indicate the Egress rate counter decrement value.</p> <p>NOTE: The rate at which the egress rate counter gets updated is still determined by the EgressRate field. This field determines the amount of decrement for each egress rate counter decrement update.</p> <p>The power on reset value for this field is 0x001 i.e., for every decrement the counter gets decremented by a value of 1.</p> <p>The expected values to be programmed for this field are: For any rate between 64kbps and 1Mbps: EgressDec = desired rate/64kbps where Egress Rate is set to 64kbps or 3906 or 0xF42.</p> <p>For any rate between 1 Mbps and 100 Mbps: EgressDec = desired rate/1 Mbps where Egress Rate is set to 1Mbps or 250 or 0xFA.</p> <p>For any rate between 100 Mbps and 1Gbps: EgressDec = desired rate /10 Mbps where Egress Rate is set to 10Mbps or 25 or 0x19.</p>

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



Link Street®

88E6321/88E6320 Functional Specification

Table 76: Egress Rate Control 2
Offset: 0x0A or Decimal 10

15:14	Count Mode	RWS to 0x2	<p>Egress rate limiting count mode. These bits are used to indicate which bytes in the transmitted frames are counted for egress rate limiting as follows:</p> <p>00 = Frame based 01 = Count all Layer 1 bytes 10 = Count all Layer 2 bytes 11 = Count all Layer 3 bytes</p> <p>Frame based: The egress rate limiting is done based on frame count as opposed to the byte count of the packet.</p> <p>Layer 1 = Preamble (8 bytes) + Frame's DA to CRC + IFG (12 bytes) + Header (if enabled – port offset 0x04) Layer 2 = Frames's DA to CRC Layer 3 = Frames's DA to CRC – 18¹ – 4 (if the frame is tagged)²</p> <p>A frame is considered tagged if the egress frame going out onto the wire is tagged.</p> <p>When Count Mode is Frame based the Frame Overhead bits (port offset 0x09) must be 0x0.</p>
13:12	Schedule	RWR	<p>Port's Scheduling mode.</p> <p>00 = Use an weighted round robin queuing scheme (default it 8, 4, 2, 1) 01 = Use Strict for priority 3 and use weighted round robin for priorities 2, 1 and 0 (default is 4, 2, 1) 10 = Use Strict for priorities 3 and 2 and use weighted round robin for priorities 1 and 0 (default is 2, 1) 11 = Use a Strict priority scheme for all priorities</p>

Table 76: Egress Rate Control 2
Offset: 0x0A or Decimal 10

11:0	Egress Rate	RWR	<p>Egress data rate shaping. The EgressRate bits modify this port's effective transmission rate together with the EgressDec bits (Egress Rate Control, offset 0x09) and the CountMode bits (above). When this register is cleared to zero egress rate limiting is disabled.</p> <p>CountMode NOT Equal to 0x0 (Layer 1, 2 or 3 bytes): The devices use the following formula to limit the Egress data rate when the CountMode is NOT equal to 0x0:</p> $\text{EgressRate} = 8 \text{ bits} * \text{Egress Dec} / (32 \text{ ns} * \text{Desired Egress Rate bits/sec})$ <p>For example: $\text{CountMode} = 0x2; \text{Desired Rate} = 640\text{kbps};$ $\text{EgressDec} = 640 \text{ kbps} / 64 \text{ kbps} = 0x0A$ $\text{EgressRate} = 8 \text{ bits} * 10 / (32\text{ns} * 640000 \text{ bits/sec})$ $= 3906 \text{ or } 0xF42$</p> <p>If CountMode is not equal to zero, the desired rate can vary from 64 kbps to 1 Gbps in the following increments: Desired rate between 64 kbps and 1 Mbps in increments of 64 kbps. Desired rate between 1 Mbps to 100 Mbps in increments of 1 Mbps. Desired rate between 100 Mbps to 1 Gbps in increments of 10 Mbps.</p> <p>CountMode Equal to 0x0 (Frame rate): The devices use the following formula to limit the Egress data rate when the CountMode is equal to 0x0:</p> $\text{EgressRate} = \text{EgressDec} / (32 \text{ ns} * \text{Desired Egress Rate frames/sec})$ <p>Where EgressDec is recommended to be programmed to a 0x01 when CountMode = 0x0.</p> <p>For example: $\text{CountMode} = 0x0; \text{Desired Rate} = 10k \text{ frames per second}$ Frame size is assumed to be 64Bytes and EgressDec is assumed to be 0x1. $\text{EgressRate} = 1 / (32\text{ns} * 10000 \text{ frames/sec})$ $= 3125 \text{ or } 0xC34$</p> <p>In CountMode = 0x0, the desired frame rate can vary from 7.6k to 1.488M frames per second.</p> <p>Egress Rate Shaping transmits a frame at wire speed counting the transmitted bytes determined by CountMode above. The value in this register determines the time it takes for the transmitted byte count to reach zero. When it reaches zero, the next frame is allowed to be transmitted and the process repeats. This burstless rate shaping is the best method for supporting the minimal amount of buffering required in the link partner this device is connected to.</p>

1. The 18 bytes are: 6 for DA, 6 for SA, 2 for EtherType and 4 for CRC.
2. Only one tag is counted even if the frame contains more than one tag (i.e. it is Provider Tagged).



Link Street®

88E6321/88E6320 Functional Specification

Table 77: Port Association Vector
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
15	HoldAt1	RWR	Hold Aging ATU Entries at an Entry State value of 1. When this bit is cleared to a zero normal Aging occurs for ATU entries associated with this port. When this bit is set to a one ATU entries associated with this port (either directly or indirectly because the entry contained a Trunk association) will age down to an Entry State of 0x1 but will not go to 0x0 (0x0 would purge the entry).
14	IntOnAgeOut	RWR	Interrupt on Age Out. When aging is enabled, all address entries in the ATU's address database are periodically aged (non-static entries have their EntryState bits decremented by 1 until they reach the value of 0x01 or 0x0). When a non-static entry is aged from an EntryState value of 0x1, and if that entry is associated with this port (either directly or indirectly because the entry contained a Trunk association), and if this IntOnAgeOut bit is set to a one, an AgeOutViolation (global 0x0B) will be captured for that entry. If the port's HoldAt1 bit (above) is zero then ATU entries will automatically age out (i.e., their EntryState will be written back to 0x0). But if the port's HoldAt1 bit is one aging entries with an EntryState = 0x1 will remain with not be automatically purged (i.e., their EntryState will remain at 0x1).
13	LockedPort	RWR	Locked Port. When this bit is cleared to a zero, normal address learning will occur. When this bit is set to a one CPU directed learning (needed for 802.1X MAC authentication) is enabled on this port. In this mode, an ATU Miss Violation interrupt occurs when a new SA address is received in a frame on this port. Automatically SA learning and refreshing is disabled in this mode. If the ATUAgeIntEn (global 2, offset 0x05) is enabled then ATU Miss Violations will also occur if a frame's SA is already in the address database, but it has an EntryState less than 0x4 (i.e., the entry is about half way aged out). Station moves will not auto refresh and will generate an ATU Miss Violation. This is done so CPU directed learning can refresh entries still being used before they age out. If RefreshLocked (below) is enabled then auto refreshing of known addresses will occur even if this port is Locked. No ATU Miss Violations from known addresses will occur either (regardless of the setting of the ATUAgeIntEn bit). This bit needs to be cleared to a zero when hardware address learn limiting is enabled on the port (Port ATU Control, offset 0x0C) so auto learning will occur before the limit is reached.
12	IgnoreWrongData	RWR	Ignore Wrong Data. All frame's SA addresses are searched for in the ATU's address database. If the frame's SA address is found in the database and if the entry is 'static' (see Section 6.8.1) or if the port is 'locked' (see bit 13 above), the source port's bit is checked to ensure the SA has been assigned to this port. If the SA is NOT assigned to this port it is considered an ATU Member Violation. If the IgnoreWrongData bit is cleared to a zero, an ATU Member Violation interrupt will be generated. If the IgnoreWrongData bit is set to a one the ATU Member Violation error is masked and ignored.

Table 77: Port Association Vector
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
11	Refresh Locked	RWR	Auto Refresh known addressed when port is Locked. Already known addresses will be auto refreshed (i.e., their Entry State will be updated to 0x7 whenever this address is used as a source address in a frame on this port) even when this port is Locked (see the LockedPort bit above) when this bit is set to a one. Station moves are not auto refreshed in this mode (i.e., the normal station move interrupt is generated if IgnoreWrongData, bit 12 above, is cleared). When this bit is cleared to a zero auto refreshing will not occur on Locked ports.
10:7	Reserved	RES	Reserved for future use.
6:0	PAV	RWS to all zeros except for this port's bit	<p>Port Association Vector for ATU learning. The value in these bits is used as the port's DPV on automatic ATU Learning or Entry_State refresh whenever these bits contain a non-zero value. When these bits are all zero, automatic Learning and Entry_State refresh is disabled on this port.</p> <p>For normal switch operation, this port's bit should be the only bit set in the vector. These bits must only be changed when frames are not entering the port (see PortState bits in Port Control – Table 70).</p> <p>The PAV bits can be used to set up port trunking (along with the VLANTable bits). For the two ports that form a trunk, set both of their port's bits in both port's PAV registers, then use the VLANTable (port offset 0x06) to isolate the two ports from each other, or to use the Trunk Mask table (Global 2 offset 0x07) to steer the traffic from the other ports down the desired trunk line of the pair using DA/SA Load Balancing.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 78: Port ATU Control¹
Offset: 0x0C or Decimal 12

Bits	Field	Type	Description
15	Read LearnCnt	RWR	<p>Read the current number of ‘active’ unicast MAC addresses associated with this port. When this bit is cleared to a zero the LearnLimit bits are accessible below. When this bit is set to a one the LearnCnt bits are accessible below.</p> <p>NOTE: Changing this bit from a 1 to a 0 will cause the bits 7:0 to be written to the LearnLimit register (i.e., a single write to this register that results with this bit being 0 will also write bit 7:0 to the LearnLimit register) unless bit 12 is set to a one. This 1 to 0 transition of this bit can be accomplished as long as bits 7:0 are written to the same value that they were prior to setting this bit to a 1². Alternately, bit 12 can be set to a one to prevent bits 7:0 from overwriting the LearnLimit register setting.</p>
14	Limit Reached	RO	<p>Limit Reached. This bit is set to a one when the port can no longer auto learn any more MAC addresses because the address learn limit set on this port has been reached.</p> <p>When this bit is set to a one the device will act as if the port is Locked (port offset 0x0B) and the SA Filtering mode is Drop on Lock (port offset 0x04). The port’s LockedPort and SAFiltering bits will not change in value, however. In fact the LockedPort and SAFiltering[0] bits must be, and stay zeros for the hardware address learn limiting to work properly. SAFiltering[1] can be zero or one allowing address learn limiting to work with the Drop On UnLock mode.</p>
13	OverLimit IntEn	RWR	Over Limit Interrupt Enable. An ATU Miss Violation will be generated when this bit is set to a one and a new source address is trying to be auto learned, but can’t, because the Limit Reached bit, above, is set. Clearing this bit to a zero will prevent ATU Miss Violations in this case.
12	KeepOldLearnLimit	RWR	Keep Old Learn Limit. When this bit is set to a zero and ReadLearnCnt (bit 15) is a zero, bits 7:0 will be written to the LearnLimit register. When this bit is set to a one, bits 7:0 will not be written to the LearnLimit register. This bit being a one allows the ReadLearnCnt bit (bit 15) to toggled without modifying the LearnLimit’s register value.
11:10	Reserved	RES	Reserved for future use.

Table 78: Port ATU Control¹
Offset: 0x0C or Decimal 12

Bits	Field	Type	Description
9:0	LearnLimit/LearnCnt	RWR/RO	<p>Port's Auto Learning Limit or port's current Auto Learning count.</p> <p>When the ReadLearnCnt bit above is cleared to zero these bits are used to enable Auto Learning limits on the port as defined below. In this mode the reading and writing of this register goes to the LearnLimit register. When the ReadLearnCnt bit is set to a one these bits are used to read back the port's current Auto Learning counter (LearnCnt). In this mode writing to these bits will have no effect (so read/modify/write operations to the ReadLearnCnt bit to toggle modes can still be done).</p> <p>When ReadLearnCnt = 0 and these bits are cleared to zero, normal address learning and frame policy occurs.</p> <p>When ReadLearnCnt = 0 and these bits are set to a non-zero value and the port is not a member of a Trunk (port offset 0x05), the number of MAC addresses that can be learned on this port are limited to the number defined in these bits. Automatic learning and frame policy will occur normally until the number of unicast MAC addresses auto-learned from this port reaches this port's LearnLimit (addresses that were learned from this port but were aged out are not counted – i.e., this register limits the number of 'active' unicast MAC addresses associated to this port). When the LearnLimit has been reached any frame that ingresses this port with a source MAC address not already in the address database that is associated with this port will be discarded (the port will act as if the port is Locked and the port's DropOnLock SAFIItering mode is set). Normal auto-learning will resume on the port as soon as the number of 'active' unicast MAC addresses associated to this port is less than the LearnLimit (due to address aging).</p> <p>When ReadLearnCnt = 1 these bits become read only and return the current number of 'active' unicast MAC addresses associated to this port.</p> <p>NOTE: The LearnCnt counter will be held at zero if the LearnLimit = 0 (i.e., whenever the limit function is disabled the LearnCnt is re-initialized). This feature will not work when this port is configured as a Trunk port (port offset 0x05). The only CPU directed ATU Operations that effect the LearnCnt counter is the ATU Flush All Entries and the ATU Flush All Non-Static Entries. In both cases the LearnCnt is cleared to zero. This means that a CPU directed ATU Load, Purge or Move of one or more unicast addresses associated with this port will not have any effect on the LearnCnt's value.</p> <p>Care is needed when enabling this feature. 1st disable learning on the ports. 2nd flush all non-static addresses in the ATU. 3rd define the desired limit for the ports. 4th re-enable learning on the ports.</p>

1. This hardware Learn Limit feature requires Learn2All must = 1 (global offset 0x0A).
2. If the LearnLimit is set to a value that is different from what it was before reading the LearnCtr unpredictable results will occur. It is best to set the LearnLimit prior to taking the port out of the Disabled or Blocking Port State (port offset 0x04). If the LearnLimit must be changed, Block the port, clear the LearnCtr (set LearnLimit to 0x00) and Move all non-Static ATU entries from this port to port 0xF (to disassociate all entries from this port - global offset 0x0B) prior to setting the new LearnLimit's value.



Link Street®

88E6321/88E6320 Functional Specification

Table 79: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
15:14	DAPri Override	RWR	<p>DA Priority Override. When these bits are cleared to a zero normal frame priority processing occurs. When either of these bits are set to a one then DA ATU priority overrides can occur on this port. A DA ATU priority override occurs when the source address of a frame results in an ATU hit where the DA's MAC address returns an EntryState that indicates Priority Override. When this occurs three forms of priority overrides are possible:</p> <p>If DAPriOverride[0] is set to a one, PRI value assigned to the frame's DA (in the ATU database) is used to overwrite the frame's previously determined FPri (frame priority). If the frame egresses tagged the priority in the frame will be this new PRI value. DA frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, port offset 0x04).</p> <p>If DAPriOverride[1] is set to a one, the two upper bits of the PRI value assigned to the frame's DA (in the ATU database) is used to overwrite the frame's previously determined QPri (queue priority). The QPri is used internally to map the frame to one of the egress queues inside the switch. QPri override will not affect the contents of the frame in any way. DA queue priority override needs to be set on DSA ports to keep the frame in the correct queue Cross-chip.</p> <p>If both DAPriOverride bits are a one then both the above overrides take place on the frame.</p> <p>The DA ATU Priority Override has highest priority over the port's Default Priority, the frame's IEEE and/or IP priorities, the frame's AVB Queue priority, the VTU Priority Override and the SA Priority Override.</p> <p>If a frame's DA is contained in the ATU with a MGMT Entry State the frame's priority will be overridden regardless of the state of this bit.</p>

Table 79: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
13:12	SAPri Override	RWR	<p>SA Priority Override. When these bits are cleared to a zero normal frame priority processing occurs. When either of these bits are set to a one then SA ATU priority overrides can occur on this port. An SA ATU priority override occurs when the source address of a frame results in an ATU hit where the SA's MAC address returns an EntryState that indicates Priority Override. When this occurs three forms of priority overrides are possible:</p> <p>If SAPriOverride[0] is set to a one, PRI value assigned to the frame's SA (in the ATU database) is used to overwrite the frame's previously determined FPri (frame priority). If the frame egresses tagged the priority in the frame will be this new PRI value. SA frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, port offset 0x04).</p> <p>If SAPriOverride[1] is set to a one, the two upper bits of the PRI value assigned to the frame's SA (in the ATU database) is used to overwrite the frame's previously determined QPri (queue priority). The QPri is used internally to map the frame to one of the egress queues inside the switch. QPri override will not affect the contents of the frame in any way. SA queue priority override needs to be set on DSA ports to keep the frame in the correct queue Cross-chip.</p> <p>If both SAPriOverride bits are a one then both the above overrides take place on the frame.</p> <p>The SA ATU Priority Override has higher priority than the port's Default Priority, the frame's IEEE and/or IP priorities, the frame's AVB Queue priority, and the VTU Priority Override. The priority determined by the frame's SA can be overridden, however, by the frame's DA Priority Override.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 79: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
11:10	VTUPri Override	RWR	<p>VTU Priority Override. When these bits are cleared to a zero normal frame priority processing occurs. When either of these bits are set to a one then VTU priority overrides can occur on this port. A VTU priority override occurs when the determined VID of a frame¹ results in a VID whose VIDPRIOverride bit in the VLAN database is set to a one. When this occurs three forms of priority overrides are possible:</p> <p>If VTUPriOverride[0] is set to a one, the VIDPRI value assigned to the frame's VID (in the VLAN database) is used to overwrite the frame's previously determined FPri (frame priority). If the frame egresses tagged the priority in the frame will be this new VIDPRI value. VID frame priority override is not recommended to be set on DSA or Ether Type DSA ports (see FrameMode, port offset 0x04).</p> <p>If VTUPriOverride[1] is set to a one, the VIDPRI value assigned to the frame's VID (in the VLAN database) is used to overwrite the frame's previously determined QPri (queue priority). The QPri is used internally to map the frame to one of the egress queues inside the switch. QPri override will not affect the contents of the frame in any way. VID queue priority override needs to be set on DSA ports to keep the frame in the correct queue cross-chip.</p> <p>If both VTUPriOverride bits are a one then both the above overrides take place on the frame.</p> <p>The VTU Priority Override has higher priority than the port's Default Priority and the frame's IEEE and/or IP priorities, and the frame's AVB Queue priority. The priority determined by the frame's VID can be overridden, however, by the frame's SA and/or DA Priority Overrides.</p>
9	Mirror SA Miss	RWR	Mirror Source Address Misses to the MirrorDest port (Global 1 offset 0x1A). When this bit is cleared to a zero normal operation occurs. When this bit is set to a one and a Source Address search miss occurs (i.e., the frame's SA is not currently in the ATU), the frame is mirrored to the port pointed to by the Mirror Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including PIRL, the Port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolMirror (Policy Mirror) frame.
8	Mirror VTU Miss	RWR	Mirror VLAN Identifier Misses to the MirrorDest port (Global 1 offset 0x1A). When this bit is cleared to a zero normal operation occurs. When this bit is set to a one and a VID search miss occurs (i.e., the VID assigned to the frame is not currently in the VTU), the frame is mirrored to the port pointed to by the Mirror Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including PIRL, the Port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolMirror (Policy Mirror) frame.

Table 79: Priority Override Register
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
7	Trap DA Miss	RWR	Trap Destination Address Misses to CPU. When this bit is cleared to a zero normal operation occurs. When this bit is set to a one and a Destination Address search miss occurs (i.e., the frame's DA is not currently in the ATU), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including PIRL, the Port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
6	Trap SA Miss	RWR	Trap Source Address Misses to CPU. When this bit is cleared to a zero normal operation occurs. When this bit is set to a one and a Source Address search miss occurs (i.e., the frame's SA is not currently in the ATU), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including PIRL, the Port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
5	Trap VTU Miss	RWR	Trap VLAN Identifier Misses to CPU. When this bit is cleared to a zero normal operation occurs. When this bit is set to a one and a VID search miss occurs (i.e., the VID assigned to the frame is not currently in the VTU), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including PIRL, the Port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
4	Trap TCAM Miss (88E6321 only)	RWR	Trap TCAM Misses to CPU. When this bit is cleared to a zero normal operation occurs. When this bit is set to a one and a TCAM search miss occurs (i.e., no TCAM entry matched the frame), the frame is mapped to the port pointed to by the CPU Dest register only (Global 1, offset 0x1A). If the frame is filtered by some other ingress policy (including PIRL, the Port Ingress Rate Limiter – Global 2, offsets 0x09 & 0x0A), it will be discarded instead. If the frame egresses a DSA port it will be marked and considered a PolTrap (Policy Trap) frame.
3:2	Reserved	RES	Reserved for future use.



Link Street®

88E6321/88E6320 Functional Specification

Table 79: Priority Override Register

Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
1:0	TCAM Mode (88E6321 only)	RWR	<p>TCAM Mode. These bits determine the TCAM's mode as follows:</p> <p>00 = TCAM disabled (default) 01 = TCAM enabled for 48 byte searches only 10 = TCAM enabled for 48 byte and/or 96 bytes searches 11 = Reserved for future use</p> <p>NOTE: Do not change the value of these bits while frames are flowing into this port. You must first put the port in the Disable Port State (Port offset 0x04), then change these TCAM Mode bits, then re-enable the port's Port State bits.</p> <p>NOTE: For proper TCAM operation the TCAM Mode must match the size of the TCAM entries that will be used on this port. Specifically, if a TCAM entry is larger than the TCAM Mode setting on this port, that TCAM entry's VID and Load Balance Overrides cannot be used.</p> <p>NOTE: TCAM compares are performed in 48 byte increments. Therefore 64 to 95 byte frame sizes can match to 48 byte TCAM entries only.</p>

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

Table 80: Policy Control Register¹
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
15:14	DA Policy	RWR	<p>DA Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then DA Policy Mapping can occur on this port. DA Policy Mapping occurs when the DA of a frame is contained in the ATU address database with an Entry State that indicates Policy (global offset 0x0C). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none"> 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>
13:12	SA Policy	RWR	<p>SA Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then SA Policy Mapping can occur on this port. SA Policy Mapping occurs when the SA of a frame is contained in the ATU address database with an Entry State that indicates Policy (global offset 0x0C). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none"> 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>
11:10	VTU Policy	RWR	<p>VTU Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then VTU Policy Mapping can occur on this port. VTU Policy Mapping occurs when the VID of a frame² is contained in the VTU database with the VidPolicy bit set to a one (global offset 0x02). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none"> 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 80: Policy Control Register¹

Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
9:8	EType Policy	RWR	<p>EType Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then EType Policy Mapping can occur on this port if the port's FrameMode is Normal Network (port offset 0x04). EType Policy Mapping occurs when the EtherType of a frame matches the PortEType register (port offset 0x0F). When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none">00 = Normal frame switching01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A)10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A)11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>
7:6	PPPoE Policy	RWR	<p>PPPoE Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then PPPoE Policy Mapping can occur on this port. PPPoE Policy Mapping occurs when the EtherType of a frame matches 0x8863. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none">00 = Normal frame switching01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1C)10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1C)11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>
5:4	VBAS Policy	RWR	<p>VBAS Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then VBAS Policy Mapping can occur on this port. VBAS Policy Mapping occurs when the EtherType of a frame matches 0x8200. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none">00 = Normal frame switching01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A)10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A)11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>

Table 80: Policy Control Register¹
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
3:2	Opt82 Policy	RWR	<p>DHCP Option 82 Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then DHCP Option 82 Policy Mapping can occur on this port. DHCP Option 82 Policy Mapping occurs when the ingressing frame is an IPv4 UDP with a UDP Destination port = 0x0043 (or decimal 67) or 0x0044 (or decimal 68) or an IPv6 UDP with a UDP Destination port = 0x0223 or 0x0222. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none"> 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>
1:0	UDP Policy	RWR	<p>UDP Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then UDP Policy Mapping can occur on this port. UDP Policy Mapping occurs when the ingressing frame is a Broadcast IPv4 UDP or a Multicast IPv6 UDP. When this occurs the mapping of non-filtered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none"> 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Discard (filter) the frame <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>

1. Policy should only be performed on Normal or Provider ports (see Frame Mode, port offset 0x04).
2. The VID of a frame could be a tagged frame's VID or the port's DefaultVID.



Link Street®

88E6321/88E6320 Functional Specification

Table 81: Port E Type
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
15:0	Port EType	RWS to 0x9100	<p>Port's Special Ether Type. This Ether Type is used for many features depending upon the mode of the port (as defined by the port's EgressMode and FrameMode bits – in Port Control, port offset 0x04).</p> <p>If the port's FrameMode is Normal Network mode, this register's value can be used to Trap, Mirror or Discard frames that ingress this port with this Ether Type (see ETypePolicy register at port offset 0x0D).</p> <p>If the port's FrameMode is Provider mode, this register's value is used as the Provider Tag Ether type added to frames that egress this port. It is also used as the expected Provider Tag Ether type on frames that ingress this port. The removal of the Provider Tags during ingress 'normalizes' the frame in memory so it can be switched to Customer ports or to another Provider Port (where it will get that port's PortEType added as the Provider Tag Ether type).</p> <p>If the port's FrameMode is Ether type Marvell® DSA Tag mode, this register's value is used as the Marvell DSA Ether type added to the appropriate frames that egress this port (either all frames or just control frames as determined by the port's EgressMode bit, offset 0x04). It is also used to match an ingressing frame's Ether type to indicate which frames contain a Marvell DSA Ether type tag. Frames that contain an Ether typed Marvell DSA Tag are 'normalized' during ingress to be stored in memory as non-Ether typed Marvell DSA tagged frames.</p>

Table 82: LED Control
Offset: 0x16 or Decimal 22

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

The individual registers accessed by the LED Control register are described in [Table 84](#) through [Table 89](#).

The LED control in the registers are organized as 2 LEDs per port for 3 ports. The physical LEDs on the device pins are organized as 3 Rows with 2 Columns. [Table 83](#) shows the port to physical mapping.

Table 83: LED Port to Physical Mapping

Bits	C0_LED	C1_LED
R0_LED	Port 3 LED 0	Port 3 LED 1
R1_LED	Port 4 LED 0	Port 4 LED 1
R2_LED	Alternate LED 0	Alternate LED 1



Note The device supports 6 LEDs or 2 LEDs per PHY port with 2 spare LEDs for the other Ports. Port 6 does not have LED Control registers (Port's 0 to 2's registers are identical to Port's 3 & 4's, but their LED signals are not brought out of the package).

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



Link Street®

88E6321/88E6320 Functional Specification

Table 84: LED 0 & 1 Control, Register Index: 0x00 of LED Control for Ports 0 to 4

Bits	Field	Type	Description
10:8	Reserved	RES	Reserved for future use.
7:4	LED1 Select	RWR or RWS based on LED Config ¹	LED 1 Selection. These bits select LED[1]'s output as follows: 0x0 = Port 5's Special LED 0x1 = 10/100 Link/Act (off = no link, on = 10 or 100 link, blink = activity) 0x2 = 10/100 Link/Act (off = no link, on = 10 or 100 link, blink = activity) 0x3 = Gig Link (off = no link, on = Gig link) 0x4 = Port 4's Special LED 0x5 = Reserved for future use 0x6 = 10/Gig Link/Act (off = no link, on = 10 or Gig link, blink = activity) 0x7 = 10/Gig Link (off = no link, on = 10 or Gig link) 0x8 = Activity (off = no link, blink on = activity) 0x9 = 100 Link (off = no link, on = 100 link) 0xA = 100 Link/Act (off = no link, on = 100 link, blink = activity) 0xB = 10/100 Link (off = no link, on = 10 or 100 link) 0xC = PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On
3:0	LED0 Select	RWR or RWS based on LED Config ²	LED 0 Selection. These bits select LED[0]'s output as follows: 0x0 = Link/Act/Speed by Blink Rate ³ (off = no link, on = link, blink = activity, blink speed = link speed ⁴) 0x1 = 100/Gig Link/Act (off = no link, on = 100 or Gig link, blink = activity) 0x2 = Gig Link/Act (off = no link, on = Gig link, blink = activity) 0x3 = Link/Act (off = no link, on = link, blink = activity) 0x4 = Port 3's Special LED 0x5 = Port 4's Special LED 0x6 = Duplex/Collision (off = half-duplex, on = full-duplex, blink = Collision) 0x7 = 10/Gig Link/Act (off = no link, on = 10 or Gig link, blink = activity) 0x8 = Link (off = no link, on = link) 0x9 = 10 Link (off = no link, on = 10 link) 0xA = 10 Link/Act (off = no link, on = 10 link, blink = activity) 0xB = 100/Gig Link (off = no link, on = 100 or Gig link) 0xC = PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On

1. This register will reset to a value of 0x0 to 0x3 based on the value of the LED_SEL[1:0] configuration pins (see the Pinlist)
2. This register will reset to a value of 0x0 to 0x3 based on the value of the LED_SEL[1:0] configuration pins (see the Pinlist)
3. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be $\frac{1}{2}$ the Gig's selected rate and the 10 blink rate is $\frac{1}{2}$ of the 100 blink rate.
4. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 3 times on each new link up, even if there is no activity, so the speed of the link can be observed.

Table 85: LED 0 & 1 Control, Register Index: 0x00 of LED Control for Alternate LEDs Port 5

Bits	Field	Type	Description
10:8	Reserved	RES	Reserved for future use.
7:4	LED1 Select	RWR or RWS based on LED Config ¹	<p>LED 1 Selection. These bits select LED[1]'s output as follows:</p> <p>0x0 = Port 1 Link/Act/Speed² by Blink Rate³ (off = no link, on = link, blink = activity, blink/speed = link speed⁴) 0x1 = Port 6 Link/Act (off = no link, on = link, blink = activity) 0x2 = Port 5 Link/Act (off = no link, on = link, blink = activity) 0x3 = Port 4 Special LED (off = no link, on = link) 0x4 = Port 5 Link/Act (off = no link, on = link, blink = activity) 0x5 = Port 6 Link (off = no link, on = link) 0x6 = Port 6 Duplex/Collision (off = half-duplex, on = full-duplex, blink = col) 0x7 = Port 6 Link/Act/Speed by Blink Rate⁵ (off = no link, on = link, blink = activity, blink speed = link speed⁶) 0x8 = Port 3's Special LED 0x9 = Port 4's Special LED 0xA = Port 5's Special LED 0xB = Reserved for future use. 0xC = Port 6 PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On</p>
3:0	LED0 Select	RWR or RWS based on LED Config ⁷	<p>LED 0 Selection. These bits select LED[0]'s output as follows:</p> <p>0x0 = Port 0 Link/Act/Speed⁸ by Blink Rate⁹ (off = no link, on = link, blink = activity, blink speed = link speed¹⁰) 0x1 = Port2 Link/Act (off = no link, on = link, blink = activity) 0x2 = Port2 Link/Act (off = no link, on = link, blink = activity) 0x3 = Port 3's Special LED 0x4 = Port 4's Special LED 0x5 = Port 5's Special LED 0x6 = Port 5 Duplex/Collision (off = half-duplex, on = full-duplex, blink = Collision) 0x7 = Port 5 Link/Act/Speed by Blink Rate¹¹ (off = no link, on = link, blink = activity, blink speed = link speed¹²) 0x8 = Port 6 Link/Act (off = no link, on = link, blink = activity) 0x9 = Port 2 Duplex/Collision (off = half duplex, on = full duplex, blink = col) 0xA = Port 2 Link/Act/Speed by Blink Rate¹³ (off = no link, on = link, blink = activity, blink speed = link speed¹⁴) 0xB = Reserved for future use 0xC = Port 5 PTP Act (blink on = PTP activity) 0xD = Force Blink 0xE = Force Off 0xF = Force On</p>



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

1. This register will reset to a value of 0x0 to 0x3 based on the value of the LED_SEL[1:0] configuration pins (see the Pinlist)
2. This mode is intended to be used with the large package only where Port 1 is connected to a SERDES.
3. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
4. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 3 times on each new link up, even if there is no activity, so the speed of the link can be observed.
5. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
6. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 3 times on each new link up, even if there is no activity, so the speed of the link can be observed.
7. This register will reset to a value of 0x0 to 0x3 based on the value of the LED_SEL[1:0] configuration pins (see the Pinlist)
8. This mode is intended to be used with the large package only where Port 0 is connected to a SERDES.
9. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
10. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 3 times on each new link up, even if there is no activity, so the speed of the link can be observed.
11. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
12. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 3 times on each new link up, even if there is no activity, so the speed of the link can be observed.
13. When this mode is chosen the Blink Rate register's contents (index 0x06) are used to control the Gig blink rate. The 100 blink rate is be ½ the Gig's selected rate and the 10 blink rate is ½ of the 100 blink rate.
14. Gig blinks at 84 mSec, 100 blinks at 170 mSec and 10 blinks at 340 mSec assuming the default register settings. The LED will blink 3 times on each new link up, even if there is no activity, so the speed of the link can be observed.

Table 86: Stretch and Blink Rate Control, Register Index: 0x06 of LED Control

Bits	Field	Type	Description
10:8	Skip Columns	RWR	<p>Skip LED Columns. LED Column 0 is always active, but LED Columns 1 can be selectively active or not. Setting the Column's bits to a one deactivates the LED Column. Clearing the Column's bit to a zero activates the LED Column. Bit 8 controls LED Column 1, bit 9 and bit 10 are used to select the LED direct drive mode.</p> <p>Setting all these bits to a one is a special case where Ports 0, 1, 3 and 4 LED 0 are directly driven on different LED/EEPROM pins without any LED multiplexing (see the pin list for which pins are assigned to which ports). In this mode the pins are active low to light a LED.</p> <p>Although the LEDs are not multiplexed in this mode, the EEPROM can still be accessed.</p> <p>NOTE: These bits only exist on Port 3.</p>
7	Reserved	RES	Reserved for future use.
6:4	Pulse Stretch	RWS to 0x1	<p>Pulse Stretch Selection for all the LEDs on this port. These bits select the port's LED stretch duration as follows:</p> <ul style="list-style-type: none"> 0x0 = no pulse stretching 0x1 = 21 mSec (Default) 0x2 = 42 mSec 0x3 = 84 mSec 0x4 = 168 mSec 0x5 to 0x7 = Reserved for future use
3	Reserved	RES	Reserved for future use.
2:0	Blink Rate	RWS to 0x2	<p>Blink Rate Selection for all the LEDs on this port. These bits select the port's LED blink rate as follows:</p> <ul style="list-style-type: none"> 0x0 = 21 mSec 0x1 = 42 mSec 0x2 = 84 mSec (Default) 0x3 = 168 mSec 0x4 = 336 mSec 0x5 = 672 mSec 0x6 to 0x7 = Reserved for future use <p>NOTE: If LED0 Select = 0x0 then these bits define the Blink Rate if the port links with a speed of Gig. If the port links to a speed of 100 then the blink time will be twice the value of this register and if the port links to a speed of 10 then the blink time will be four times the value of this register.</p>



Link Street®

88E6321/88E6320 Functional Specification

**Note**

The Special LED function is different per port and currently only exist for Ports 3 to 5.

Table 87: Port 3 Special Control, Register Index: 0x07 of LED Control on Port 3

Bits	Field	Type	Description
10:7	Reserved	RES	Reserved for future use.
6:0	LAN Link/Act	RWS to 0x20	LAN Link Activity LED. Port 3's Special LED is a single Link Activity LED that is a combination of any of the selected port's Link Activity LED. If any of the selected port's Link is active this LED is on. If any of the selected Link'ed port's Activity is active this LED will blink off. This LED will be off if all the selected port's Links are down. This can be used as a front panel LED to indicate Link/Activity for any of the LAN ports. The default value selects Link/Activity for Port 5 only. The bits of this register are used to define which ports are to be considered LAN ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a LAN port for purposes of this LED only.

Table 88: Port 4 Special Control, Register Index: 0x07 of LED Control on Port 4

Bits	Field	Type	Description
10:7	Reserved	RES	Reserved for future use.
6:0	WAN Link/Act	RWS to 0x40	WAN Link Activity LED. Port 4's Special LED is a single Link Activity LED that is a combination of any of the selected port's Link Activity LED. If any of the selected port's Link is active this LED is on. If any of the selected Link'ed port's Activity is active this LED will blink off. This LED will be off if all the selected port's Links are down. This can be used as a front panel LED to indicate Link/Activity for any of the WAN ports. The default value selects Link/Activity for Port 6 only. The bits of this register are used to define which ports are to be considered WAN ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a LAN port for purposes of this LED only.

Table 89: Port 5 Special Control, Register Index: 0x07 of LED Control on Port 5

Bits	Field	Type	Description
10:7	Reserved	RES	Reserved for future use.
6:0	PTP Activity	RWS to 0x01	<p>WAN Link Activity LED. Port 5's Special LED is a single Link Activity LED that is a combination of any of the selected port's Link Activity LED. If any of the selected port's Link is active this LED is on. If any of the selected Link'ed port's Activity is active this LED will blink off. This LED will be off if all the selected port's Links are down.</p> <p>This can be used as a front panel LED to indicate Link/Activity for any of the WAN ports.</p> <p>The bits of this register are used to define which ports are to be considered WAN ports. Bit 0 is for Port 0, bit 1 is for Port 1, etc. Setting a port's bit to a one selects that port as a LAN port for purposes of this LED only.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 90: Port IEEE Priority Remapping Registers
Offset: 0x18 or Decimal 24

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:12	TagRemap3	RWS to 0x3	Tag Remap 3. All IEEE tagged frames with a priority of 3 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
11	Reserved	RES	Reserved for future use.
10:8	TagRemap2	RWS to 0x2	Tag Remap 2. All IEEE tagged frames with a priority of 2 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
7	Reserved	RES	Reserved for future use.
6:4	TagRemap1	RWS to 0x1	Tag Remap 1. All IEEE tagged frames with a priority of 1 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
3	Reserved	RES	Reserved for future use.
2:0	TagRemap0	RWR	Tag Remap 0. All IEEE tagged frames with a priority of 0 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.

Table 91: Port IEEE Priority Remapping Registers
Offset: 0x19 or Decimal 25

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:12	TagRemap7	RWS to 0x7	Tag Remap 7. All IEEE tagged frames with a priority of 7 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
11	Reserved	RES	Reserved for future use.
10:8	TagRemap6	RWS to 0x6	Tag Remap 6. All IEEE tagged frames with a priority of 6 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
7	Reserved	RES	Reserved for future use.
6:4	TagRemap5	RWS to 0x5	Tag Remap 5. All IEEE tagged frames with a priority of 5 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.
3	Reserved	RES	Reserved for future use.
2:0	TagRemap4	RWS to 0x4	Tag Remap 4. All IEEE tagged frames with a priority of 4 get this register's value as the frame's new priority inside the switch. If a tagged frame egresses the switch tagged, this new priority is written to the frame's tag.

Table 92: Queue Counter Registers
Offset: 0x1B or Decimal 27

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



Link Street®

88E6321/88E6320 Functional Specification

Table 93: Debug Counter
Offset: 0x1E or Decimal 30

Bits	Field	Type	Description
15:8	RxBad Frames/ Tx Collisions	RO	<p>Bad Counter. When CtrMode is cleared to a zero (Global 1 offset 0x1C) this counter increments each time a frame enters this port that was an error on the wire. It does not matter if the frame's CRC is fixed by ForceGoodFCS (Port offset 0x08) being set to a one, this counter will still increment. A CRC error frame is one that is 64 bytes to MaxFrameSize (Global 1, offset 0x04) with a bad CRC (including alignment errors but not dribbles). Fragments and properly formed frames are not counted.</p> <p>When CtrMode is set to a one this counter increments each time a transmit collision occurs on this port.</p> <p>The counter will wrap around back to zero. The only time this counter will not increment is when this port is Disabled (see PortState in Port offset 0x04). This register can be cleared by changing the state of the CtrMode bit in Global Control 2 (Global 1 offset 0x1C) or by a Flush All Counters for this port or all ports StatsOp command (section 1.3.2.23).</p>
7:0	RxGood Frames/Tx Transmit Frames	RO	<p>Good Counter. When CtrMode is cleared to a zero (Global 1 offset 0x1C) this counter increments each time a frame enters this port that was not an error frame on the wire. It does not matter if the frame was filtered or discarded, only that the frame was received as good on the wire (i.e., its wire size is in the range of 64 bytes to MaxFrameSize (Global 1, offset 0x04) and its CRC was good).</p> <p>When CtrMode is set to a one this counter increments each time a frame is transmitted out this port.</p> <p>The counter will wrap around back to zero. The only time this counter will not increment is when this port is Disabled (see PortState in Port offset 0x04). This register can be cleared by changing the state of the CtrMode bit in Global Control 2 (Global 1 offset 0x1C) or by a Flush All Counters for this port or all ports StatsOp command (section 1.3.2.23).</p>

Table 94: Cut Through Register
Offset: 0x1F or Decimal 31

Bits	Field	Type	Description
15:12	Enable Select (valid on Port 2, 5, and 6 only)	RWS	Port Enable Select. These bits are used to select the Px_ENABLE from one of the GPIO pins. It is the responsibility of the user to insure that the selected GPIO pin is enabled to be a GPIO pin (see GPIO Configuration register, Global 2 Scratch & Misc Index 0x60 & 0x61), and that the GPIO pin is configured to be an input (see GPIO Direction register, Global 2 Scratch & Misc Index 0x62 & 0x63). Enable Select works as follows: 0x0 to 0xE = This port's Px_ENABLE connects to GPIO[Enable Select] ¹ 0xF = This port's Px_ENABLE connects to a one (enabled) ²
11:9	Reserved	RES	Reserved for future use.
8	Cut Through (88E6321 only)	RWR	Cut Through enable. When this bit is cleared to a zero, cut through from this Ingress port cannot occur. When this bit is set to a one, frames are allowed to cut through from this Ingress port to any port and queue whose Cut Through Queue bit (below) is set to a one. Many other conditions are required before a frame can actually be cut through the switch fabric, bypassing the normal Store and Forward Queue Controller. Some (but not all) of these conditions include: Both ports must be in the Forwarding Port State (Port offset 0x04), Ingress and Egress ports must be at the same speeds and both must be in full duplex mode (Port offset 0x00). The Egress MAC must not be transmitting and the target priority queue must be empty. NOTE: If Cut Through is enabled on a port that port should not enable IPv6 Snooping (Port offset 0x04) as the IPv6 Snoop point may be after byte 64.
7:4	Reserved	RES	Reserved for future use.
3:0	Cut Through Queue (88E6321 only)	RWR	Cut Through Queues. When all of these bits are cleared to zero, cut through to this Egress port cannot occur. When any one of these bits are set to a one, frames are allowed to cut through to this Egress port, but only if the frame came from a Cut Through enabled Ingress port (see Cut Through bit above) and the frame is mapped to a priority queue whose Cut Through Queue register bit is set to a one. Bit 0 of this registers is the enable for Pri 0, bit 1 for Pri 1, etc.

1. An EnableSelect value of 0x0 selects GPIO[0], a value of 0x1 selects GPIO[1], a value of 0xE selects GPIO[14], etc.
2. The default value of 0xF sets the port's Px_ENABLE to one enabling the link on the port and drives the port's output pins. But if the port's Px_MODE (see the port's Px_OUTD pin descriptions) or C_Mode (Port offset 0x00) = 0x6 the port will be disabled overriding the port's Px_ENABLE.

10.4.1 Switch Global 1 Registers

The devices contain global registers that affect all Ethernet ports in the device. Each global register is 16-bits wide. Global registers' bit assignments are shown in Figure 64.

Figure 64: Global 1 Register Bit Map

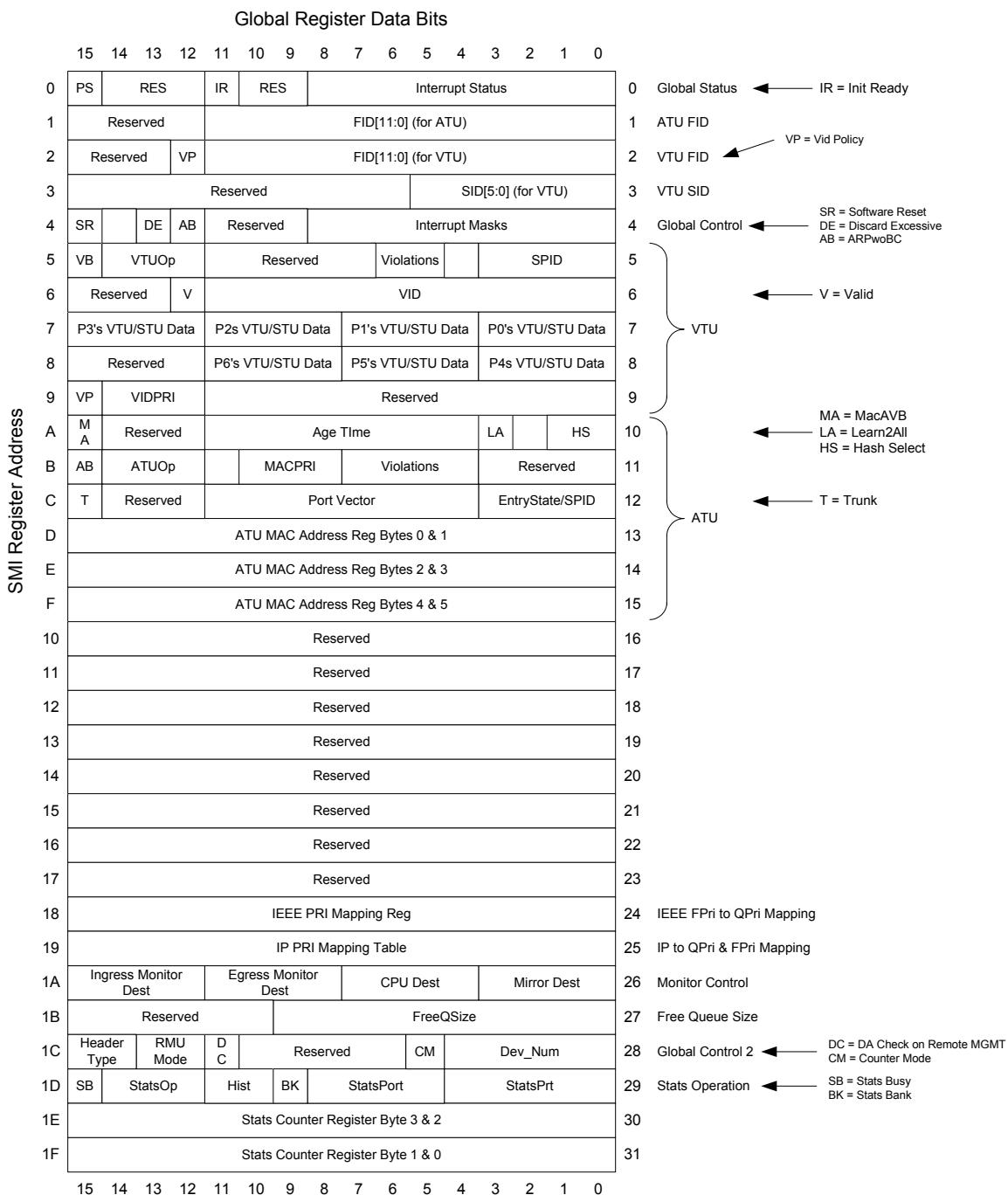


Table 95: Switch Global Status Register
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	PPUState	RO	PHY Polling Unit State. These bits indicate the state of the PPU as follows: 0 = PPU is Active detecting and initializing external PHYs. The PortStatus registers (Table 65) must not be written by software. 1 = PPU Polling. This indicates the PPU is Active polling the external PHYs. Software can write to the PortStatus registers (Table 65).
14:12	Reserved	RES	Reserved for future use.
11	InitReady	RO	SwitchReady. This bit is set to a one when the Address Translation Unit, the VLAN Translation Unit, the Queue Controller and the Statistics Controller complete their initialization and are ready to accept frames.
10:9	Reserved	RES	Reserved for future use.
8	AVBInt	RO	AVB Interrupt. If any of the per-port PTPInt bits (shown in PTP Global Status Data Structure) are set then this bit gets set. After reading the appropriate PTP Global Status and PTP Port status registers in the PTP registers space, CPU clears the PTPInt bits which in turns clears this bit. This bit being high will cause the 88E6350R/88E6350/88E6351 device's INTn pin to go low if the AVBIntEn bit in Global Control (Table 99) is set to a one.
7	DeviceInt	RO	Device Interrupt. This bit is set to a one when any of the device interrupts have at lease one active interrupt. The device interrupts are defined in the Interrupt Source register (Global 2, offset 0x00). This bit being high will cause the device's INTn pin to go low if the DevIntEn bit in Global Control (global offset 0x04) is set to a one.
6	StatsDone	LH	Statistics Done Interrupt. This bit is set to a one whenever the STATBusy bit (Table 118) transitions from a one to a zero. It is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the STATDoneIntEn bit in the Global Control register (Table 99) is set to a one.
5	VTUProb	RO	VLAN Table Problem/Violation Interrupt. This bit is set to a one if a VLAN Violation is detected. It is automatically cleared when all of the pending VTU Violations have been serviced by the VTU Get/Clear Violation Data operation (Table 100). This bit being high causes the device's INTn pin to go low if the VTUProbIntEn bit in Global Control (Table 99) is set to a one.
4	VTUDone	LH	VTU Done Interrupt. This bit is set to a one whenever the VTUBusy bit (Table 100) transitions from a one to a zero. It is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the VTUDoneIntEn bit in Global Control (Table 99) were set to a one.
3	ATUProb	RO	ATU Problem/Violation Interrupt. This bit is set to a one if the ATU cannot load or learn a new mapping due to all the available locations for an address being static or if an ATU Violation is detected. It is automatically cleared when all the pending ATU Violations have been serviced by the ATU Get/Clear Violation Data operation (Table 108). This bit being high causes the device's INTn pin to go low if the ATUProbIntEn bit in Global Control (Table 100) is set to a one.



Link Street®

88E6321/88E6320 Functional Specification

Table 95: Switch Global Status Register

Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
2	ATUDone	LH	ATU Done Interrupt. This bit is set to a one whenever the ATUBusy bit (Table 108) transitions from a one to a zero. It is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the ATUDoneIntEn bit in Global Control (Table 99) is set to a one.
1	TCAM Int (88E6321 only)	ROC	TCAM Interrupt. This bit is set to a one whenever the TCAM gets a hit where the Action Int bit is set to a one (Global 3 Page 2 offset 0x02). It is automatically cleared when read. This bit being high will cause the device's INTn pin to go low if the TCAMIntEn bit in Global Control (section 1.3.2.5) is set to a one.
0	EEInt	LH	EEPROM Done Interrupt. This bit is set to a one after the EEPROM is done loading registers or when an EEPROM operation is done (see EEPROM Control, Global 2 offset 0x14) and it is automatically cleared when read. This bit being high causes the device's INTn pin to go low if the EEIntEn bit in Global Control (Table 99) is set to a one.

Table 96: ATU FID Register

Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:0	FID	RWR	ATU MAC Address Forwarding Information Database number. If multiple address databases are not being used these bits must remain zero. If multiple address databases are being used these bits are used to set the desired address database number that is to be used on the Database supported commands (ATUOps 0x3, 0x4, 0x5 and 0x6 above). On Get/Clear Violation Data ATUOps these bits return the FID] value associated with the ATU violation that was just serviced.

Table 97: VTU FID Register
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15:13	Reserved	RES	Reserved for future use.
12	VIDPolicy	RWR	VID Policy. This bit is used to indicate any frames associated with this VID value are to be trapped to the TrapDest port (global offset 0x1A), monitored to the MirrorDest port (global offset 0x1A) or discarded. The action that takes place is determined by the frame's ingressing port's VTUPolicy bits (port offset 0xE).
11:0	FID	RWR	VTU MAC Address Forwarding Information Database (FID) number. On VTU Load and VTU GetNext operations, this field is VTU FID and it is used to separate MAC address databases by a frame's VID. If multiple address databases are not being used these bits must remain zero. If multiple address databases are being used these bits are used to set the desired address database number that is associated with a VID value on Load operations (or these bits are used to return the currently assigned FID value found in the VTU on Get Next operations).

Table 98: VTU SID Register
Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:6	Reserved	RES	Reserved for future use.
5:0	SID	RWR	VTU 802.1s Port State Information Database (SID) number. On VTU Load and VTU GetNext operations this field is the SID data that is associated with the VID that is being loaded or read in the VTU. If 802.1s multiple spanning trees are not being used these SID bits must remain zero. If multiple spanning trees are being used these bits are used to define the desired 802.1s information database (SID) number that is associated with the VID value on Load operations (or these bits are used to return the currently assigned SID value found in the VTU on Get Next operations). On STU Load and STU GetNext operations this field is used as the SID that is associated with the STU data (Global 1, offsets 0x07 to 0x09).



Link Street®

88E6321/88E6320 Functional Specification

Table 99: Switch Global Control Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15	SWReset	SC	Switch Software Reset. Writing a one to this bit causes the QC, the MAC state machines in the switch to be reset. Register values are not modified. The EEPROM is not re-read. The ATU, VTU, MIBs, PHYs are not affected by this bit. When the reset operation is complete, this bit is cleared to a zero automatically. The reset occurs immediately. To prevent transmission of CRC frames, set all of the ports to the Disabled state (Table 70), and wait for 2 ms. (i.e., the time for a maximum frame to be transmitted at 10 Mbps) before setting the SWReset bit to a one.
14	Reserved	RES	Reserved for future use.
13	Discard Excessive	RWR	Discard frames with Excessive Collisions. When this bit is set to a one frames that encounter 16 consecutive collisions are discarded. When this bit is cleared to a zero Egress frames are never discarded and the backoff range is reset after 16 consecutive collisions on a singe frame.
12	ARPwo BC	RWR	ARP detection without Broadcast checking When enabled the switch core does not check for a Broadcast MAC address as part of the ARP frame detection. It only checks the Ether Type (0x0806) and makes the decision. When disabled the switch core checks for both the Ether Type and Broadcast MAC address for ARP frame detection.
11:10	Reserved	RES	Reserved for future use.
9	Reserved	RES	Reserved for future use. NOTE: This bit use to be used for reloading of the registers using the EEPROM. This is now accomplished by another method (See EEPROM Command, Global 2 offset 0x14).
8	AVBIntEn	RO	AVB Interrupt Enable. This bit must be set to a one to allow active interrupts enabled in AVB registers in PTP Global Status Data Structure to drive the device's INTn pin low.
7	DevIntEn	RWR	Device Interrupt Enable. This bit must be set to a one to allow the Device interrupt to drive the device's INTn pin low.
6	StatsDone IntEn	RWR	Statistics Operation Done Interrupt Enable. This bit must be set to a one to allow the Stat Done interrupt to drive the device's INTn pin low.
5	VTUProb IntEn	RWR	VLAN Problem/Violation Interrupt Enable. This bit must be set to a one to allow the VTUProblem interrupt to drive the device's INTn pin low.
4	VTUDone IntEn	RWR	VLAN Table Operation Done Interrupt Enable. This bit must be set to a one to allow the VTUDone interrupt to drive the device's INTn pin low.
3	ATUProb IntEn	RWR	ATU Problem/Violation Interrupt Enable. This bit must be set to a one to allow the ATU Problem interrupt to drive the device's INTn pin low.
2	ATUDone IntEn	RWR	ATU Operation Done Interrupt Enable. This bit must be set to a one to allow the ATU Done interrupt to drive the device's INTn pin low.

Table 99: Switch Global Control Register (Continued)
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
1	TCAM IntEn (88E6321 only)	RWR	TCAM Int Interrupt Enable. This bit must be set to a one to allow the TCAMInt interrupt to drive the Agate's INTn pin low.
0	EEIntEn	RWS	EEPROM Done Interrupt Enable. This bit must be set to a one to allow the EEPROM Done interrupt to drive the device's INTn pin low.



Link Street®

88E6321/88E6320 Functional Specification

Table 100: VTU Operation Register
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	VTUBusy	SC	VLAN Table Unit Busy. This bit must be set to a one to start a VTU operation (see VTUOp below). Only one VTU operation can be executing at one time so this bit must be zero before setting it to a one. When the requested VTU operation completes, this bit will automatically be cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (Table 99).
14:12	VTUOp	RWR	VLAN Table Unit Table Opcode. The devices support the following VTU operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Flush All Entries in the VTU and STU 010 = No Operation 011 = VTU Load ¹ or Purge ² an Entry 100 = VTU Get Next ³ 101 = STU Load ⁴ or Purge ⁵ an Entry 110 = STU Get Next ⁶ 111 = Get/Clear Violation Data ⁷
11:7	Reserved	RES	Reserved for future use
6	Member Violation	RO	Source Member Violation. On Get/Clear Violation Data VTUOps, this bit is returned set to a one if the Violation being serviced is due to an 802.1Q Member Violation. A Member Violation occurs when an 802.1Q enabled Ingress port accesses the VTU with a VID that is contained in the VTU but whose Membership list does not include this Ingress port. Only the first Member Violation or Miss Violation (below) will be saved until cleared.
5	Miss Violation	RO	VTU Miss Violation. On Get/Clear Violation Data VTUOps this bit is returned set to a one if the Violation being serviced was due to an 802.1Q Miss Violation. A Miss Violation occurs when an 802.1Q enabled Ingress port accesses the VTU with a VID that is not contained in the VTU. Only the first Miss Violation or Member Violation (above) is saved until cleared.
4	Reserved	RES	Reserved for future use
3:0	SPID	RO	On the Get Violation Data VTUOp, this field returns the Source Port ID of the port that caused the violation. If SPID 0xF the source of the violations was the CPU register interface (i.e., the VTU was full during a CPU Load operation).

1. A VTU Entry is Loaded when the Valid bit (in the VTU VID register at global offset 0x06) is set to a one. This VTU Load is the only VTUOp that uses the FID & SID field and it uses them as data to be loaded along with the desired VID and its port member data.
2. An VTU Entry is Purged when the Valid bit (in the VTU VID register at global offset 0x06) is cleared to a zero.
3. A VTU Get Next operation finds the next higher VID currently in the VTU's database. The VID value (Table 101) is used as the VID to start from. To find the lowest VID set the VID field to ones. When the operation is done the VID field contains the next higher VID currently active in the VTU. To find the next VID simply issue the VTU Get Next opcode again. If the VID field is returned set to all one's with the Valid bit cleared to zero, no higher VID's were found. To Search for a particular VID, perform a VTU Get Next operation using a VID field with a value one less than the one being searched for.
4. A SID Entry is Loaded if the Valid bit (in the VTU VID register at global offset 0x06) is set to a one. This STU Load uses the SID as a pointer into the SID Translation Unit (STU). The data loaded into the STU is the lower two bits of each port's VTU Data that are used to define the 802.1s port states that are to be associated with this SID.
5. A SID Entry is Purged if it exists and the Valid bit (in the VTU VID register at global offset 0x06) is cleared to a zero.
6. A STU Get Next operation finds the next higher SID currently in the STU's database. The SID value is used as the SID to start from. To find the lowest SID set the SID field to ones. When the operation is done the SID field contains the next higher SID currently active in the STU. To find the next SID simply issue the STU Get Next opcode again. If the SID field is returned set to all one's with a Valid bit cleared to zero, no higher SID's were found. To Search for a particular SID, perform a STU Get Next operation using a SID field with a value one less than the one being searched for.
7. When the VTUProb bits is set to a one (Global Status—Table 95) the Get/Clear Violation VTUOp can be used to retrieve the data associated with the Violation. It will return the source port of the violation in the SPID field of this registers (bits 3:0) and it will return the VID of the violation in the VID field of the VTU VID register (Table 101). When all Violations currently pending in the VTU have been serviced the VTUProb bit in Global Status will be cleared to a zero.

Table 101: VTU VID Register
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description
15:13	Reserved	RES	Reserved for future use.
12	Valid	RWR	Entry's Valid bit. At the end of VTU (or STU) Get Next operations, if this bit is set to a one it indicates the VID (or SID) value below is valid (or the SID value above is valid). If this bit is cleared to a zero and the VID (or SID) is all ones, it indicates the end of the VID (or SID) list was reached with no new valid entries found.
11:0	VID	RWR	On Load or Purge operations, this bit indicates the desired operation of a Load (when set to a one) or a Purge (when cleared to a zero).
			VLAN Identifier. This VID is used in the VTU Load or VTU GetNext operation and it is the VID that is associated with the VTU data below (Table 102) or the VID that caused the VTU Violation.

Table 102: VTU/STU Data Register Ports 0 to 3 for VTU Operations
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
13:12	Member TagP3	RWR	Membership and Egress Tagging for Port 3. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP0 below.
11:10	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
9:8	Member TagP2	RWR	Membership and Egress Tagging for Port 2. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP0 below.
7:6	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
5:4	Member TagP1	RWR	Membership and Egress Tagging for Port 1. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP0 below.
3:2	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
1:0	Member TagP0	RWR	Membership and Egress Tagging for Port 0. These bits are used to support 802.1Q membership and Egress Tagging as follows: 00 = Port is a member of this VLAN and frames are to egress unmodified. 01 = Port is a member of this VLAN and frames are to egress Untagged. 10 = Port is a member of this VLAN and frames are to egress Tagged. 11 = Port is not a member of this VLAN. Any frames with this VID ¹ are discarded at ingress and are not allowed to egress this port.

1. The VID used comes from the VID in Tagged frames or the default VID assigned to Untagged frames.



Link Street®

88E6321/88E6320 Functional Specification

Table 103: VTU/STU Data Register Ports 0 to 3 for STU Operations
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:14	PortState P3	RWR	Per VLAN Port States for Port 3. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP0 below.
13:12	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
11:10	PortState P2	RWR	Per VLAN Port States for Port 2. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP0 below.
9:8	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
7:6	PortState P1	RES	Per VLAN Port States for Port 1. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP0 below.
5:4	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
3:2	PortState P0	RES	Per VLAN Port States for Port 0. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. The Per VLAN Port States are: 00 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 01 = Blocking/Listening Port State for this port for frames with this VID. 10 = Learning Port State for this port for frames with this VID. 11 = Forwarding Port State for this port for frames with this VID.
1:0	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.

Table 104: VTU/STU Data Register Ports 4 to 5 for VTU Operations
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15:10	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
9:8	Member TagP6	RWR	Ingress and Egress Membership and Egress Tagging for Port 6. These bits are used to support 802.1Q Egress membership and Egress Tagging. See MemberTagP4 below.
7:6	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations
5:4	Member TagP5	RWR	Membership and Egress Tagging for Port 5. These bits are used to support 802.1Q membership and Egress Tagging. See MemberTagP4 below.
3:2	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.
1:0	Member TagP4	RWR	Membership and Egress Tagging for Port 4. These bits are used to support 802.1Q membership and Egress Tagging as follows: 00 = Port is a member of this VLAN and frames are to egress unmodified. 01 = Port is a member of this VLAN and frames are to egress Untagged. 10 = Port is a member of this VLAN and frames are to egress Tagged. 11 = Port is not a member of this VLAN. Any frames with this VID ¹ are discarded at ingress and are not allowed to egress this port.

1. The VID used comes from the VID in Tagged frames or the default VID assigned to Untagged frames.



Link Street®

88E6321/88E6320 Functional Specification

Table 105: VTU/STU Data Register Ports 4 to 5 for STU Operations
Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:10	PortState P6	RWR	Per VLAN Port States for Port 6. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP4 below.
9:8	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations
7:6	PortState P5	RWR	Per VLAN Port States for Port 5. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. See PortStateP4 below.
5:4	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.
3:2	PortState P4	RES	Per VLAN Port States for Port 4. These bits are used to support 802.1s (per VLAN Spanning Tree) and should be cleared to zero if 802.1s is not used. The Per VLAN Port States are: 00 = 802.1s Disabled. Use non-VLAN Port States for this port for frames with this VID. 01 = Blocking/Listening Port State for this port for frames with this VID. 10 = Learning Port State for this port for frames with this VID. 11 = Forwarding Port State for this port for frames with this VID.
1:0	Reserved	RES	Reserved for future use. Will return 0x0 on VTU GetNext Operations.

Table 106: VTU/STU Data Register for VTU Operations
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15	VIDPRI Override	RWR	VID Priority Override. When this bit is set to a one the VIDPRI bits (below) are used to override the priority on any frame associated with this VID.
14:12	VIDPRI	RWR	VID Priority bits. These bits are used to override the priority on any frames associated with this VID value, if the VIDPRIOverride bit (above) is set to a one.
11:0	Reserved	RES	Reserved for future use. Will return 0x0 on STU GetNext Operations.

Table 107: ATU Control Register
Offset: 0x0A or Decimal 10

Bits	Field	Type	Description
15	MACAVB	RWR	ATU MAC entry in AVB mode. When 0x1, the ATU entries operate in AVB mode: Entry state 101x = Static AVB Unicast entry and Entry state x101 = Static AVB Multicast entry when 0x0, the ATU entries operate in non-AVB mode: Entry state 101x = Static NRL Unicast entry and Entry state x101 = Static NRL Multicast entry
14:12	Reserved	RES	Reserved for future use.
11:4	AgeTime	RWS to 0x16	ATU Age Time. These bits determine the time that each ATU Entry remains valid in the database, since its last access as a source address, before being purged. The value in this register times 15 is the age time in seconds. For example: The default value of 0x16 is 22 decimal. $22 \times 15 = 330$ seconds or 5.5 minutes, which results in an average age time of 306 seconds, or about 5 minutes (as register setting sets the maximum time with the minimum time being the Max. Time - Max. Time/7, and the average time being the average between those two times). The minimum age time is 0x1 or 15 seconds. The maximum age time is 0xFF or 3825 seconds or almost 64 minutes. If the AgeTime is set to 0x0 the Aging function is disabled and all learned addresses will remain in the database forever.
3	Learn2All	RWR	Learn to All devices in a Switch. When more than one Marvell® device is used to for a single 'switch' it may be desirable for all devices in the 'switch' to learn any address this device learns ¹ . When this bit is set to a one all other devices in the 'switch' learn the same addresses this device learns. When this bit is cleared to a zero only the devices that actually receive frames will learn from those frames. This mode typically supports more active MAC addresses at one time as each device in the switch does not need to learn addresses it may never use. Learn2All must be set to a 1 when hardware learn limiting is enabled on any port in the device (port offset 0x0C).
2	Reserved	RES	Reserved for future use.
1:0	HashSel	RWR to 0x1	Hash Select. These bits select how a MAC addressed is hashed into the ATU. 00 = Reserved 01 = Default 10 = Reserved 11 = Direct Method (no hash). Should be used for test purposes only.

1. Learn2All message learning frames will be sent out a port if that port's MessagePort bit is set to a one (Table 71). If this frame is used it is recommended that all DSA Tag ports, except for the CPU's port, have their MessagePort bit set to a one. Ports that are not DSA Tag ports (i.e., normal Network ports) should not have their MessagePort bit set to a one.



Link Street®

88E6321/88E6320 Functional Specification

Table 108: ATU Operation Register
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
15	ATUBusy	SC	Address Translation Unit Busy. This bit must be set to a one to start an ATU operation (see ATUOp below). Only one ATU operation can be executing at one time so this bit must be zero before setting it to a one. When the requested ATU operation completes, this bit is automatically be cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (Table 99).
14:12	ATUOp	RWR	Address Translation Unit Opcode. The devices support the following ATU operations. (All of these operations can be executed while frames are passing through the switch): 000 = No Operation 001 = Flush ¹ All Entries 010 = Flush all Non-Static ² Entries 011 = Load ³ or Purge ⁴ an Entry in a particular FID Database 100 = Get Next ⁵ from a particular FID Database 101 = Flush All Entries in a particular FID Database 110 = Flush all Non-Static Entries in a particular FID Database 111 = Get/Clear Violation Data ⁶
11	Reserved	RES	Reserved for future use.
10:8	MACPri	RWR	MAC Priority bits. These bits are used to override the priority on any frames associated with this MAC value, if the EntryState bits indicate MAC Priority can be used – see Section 6.8.1) and the port's SA and/or DA priority overrides are enabled (in Port Control 2 – Table 74).
7	AgeOut Violation	RO	Age Out Violation. On Get/Clear Violation Data ATUOps this bit is returned set to a one if the Violation being serviced was due to a non-static entry being aged with an EntryState = 0x1. AgeOutViolations will only occur on entries that are associated with ports whose IntOnAgeOut bit is set to a one (port offset 0x0B). Up to 2 Age Out Violations will be saved per device until cleared. An Age Out Violation will return the violating MAC in global registers at offset 0x0D, 0x0E and 0x0F. The ATU Data Register at global offset 0x0C will contain the violating MAC's Trunk bit, its DPV or Trunk ID and its Entry State. The violating MAC's PRI bits will be updated in MACPri (global offset 0x0B) and its BIN will be updated in global offset 0x06).
6	Member Violation	RO	Source Port Violation. On Get/Clear Violation Data ATUOps this bit is returned set to a one if the Violation being serviced is due to a Source Address look-up that resulted in a Hit but where the ATUData[8:0] bits does not contain the frame's Ingress port bit set to a one (i.e., a station move occurred). This violation can be masked on a per port basis by setting the port's IgnoreWrongData bit. Only the first Member Violation, Miss Violation (below) or Full Violation (below) is be saved per port until cleared.

Table 108: ATU Operation Register
Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
5	Miss Violation	RO	<p>ATU Miss Violation. On Get/Clear Violation Data ATUOps this bit is returned set to a one if the Violation being serviced is due to a Source Address look-up that resulted in a Miss on ports that are Locked (i.e., CPU directed learning is enabled on the port).</p> <p>If Age Violations are enabled (ATUAgeIntEn = 1 in global 2, offset 0x05) and Locked ports are not allowed to self refresh addresses (RefreshLocked = 0 in port offset 0x0B) this Miss Violation will also occur if the frame's Source Address was found in the address database with an EntryState less than 0x4 (i.e., it is about half way aged out).</p> <p>Only the first Miss Violation, Member Violation (above) or Full Violation (below) is saved per port until cleared.</p>
4	ATUFull Violation	RO	ATU Full Violation. On Get/Clear Violation Data ATUOps this bit is set to a one if the Violation being serviced is due to a Load ATUOp or automatic learn that could not store the desired entry. This only occurs if all available locations for the desired address contain other MAC addresses that are loaded Static. Only the first Full Violation, Member Violation (above) or Miss Violation (above) is saved per port until cleared.
3:0	Reserved	RES	Reserved for future use.

1. A Flush occurs when the EntryState ([Table 109](#)) is zero.
2. A Non-Static entry is any unicast address with an EntryState less than 0x8. All unicast frames flood until new addresses are learned.
3. An Entry is Loaded when the EntryState ([Table 109](#)) is non-zero.
4. An Entry is Purged when the EntryState ([Table 109](#)) is zero.
5. A Get Next operation finds the next higher MAC address currently in a particular ATU database (defined by the FID field - Global offset 0x01). The ATUByte[5:0] values ([Table 110](#)) are used as the starting address. To find the lowest MAC address set ATUByte[5:0] to ones. When the operation is done, ATUByte[5:0] contains the next higher MAC address. To find the next address, simply issue the Get Next opcode again. If ATUByte[5:0] is returned set to all one's with an EntryState of 0x0, no higher MAC address was found. If ATUByte[5:0] is returned set to all one's with a non-zero EntryState, the highest MAC address was found (i.e., the Broadcast address) and the end of the table was reached. To search for a particular address, perform a Get Next operation using a MAC address with a value one less than the one being searched for.
6. When the ATUProb bit is set to a one (Global Status - [Table 95](#)), the Get/Clear Violation ATUOp can be used to retrieved the data associated with the violation. When all violations currently pending in the ATU have been serviced the ATUProb bit in the Global Status is cleared to a zero.



Link Street®

88E6321/88E6320 Functional Specification

Table 109: ATU Data Register
Offset: 0x0C or Decimal 12

Bits	Field	Type	Description
15	Trunk	RWR	Trunk Mapped Address. When this bit is set to a one the data bits 7:4 below (PortVec bits [3:0]) is the Trunk ID assigned to this address. PortVec bits [10:4] must be written as zero when this bit is set to a one. When this bit is cleared to a zero the data in bits 9:4 below (PortVec bit[5:0]) is the port vector assigned to this address.
14:12	Reserved	RES	Reserved for future use.
11:4	PortVec/ ToPort & FromPort	RWR	Port Vector. If the Trunk bit, above, is zero, these bits are used as the input Port Vector for ATU Load operations and it's the resulting Port Vector from ATU Get Next operations. The lower four bits (7:4) are used as the FromPort and the next higher four bits (11:8) are used as the ToPort during move operations. If the ToPort = 0xF, the operation becomes a RemovePort (i.e., the FromPort is removed from the database and the entry is purged if the resulting PortVec equals zeros).
3:0	EntryState/ SPID	RWR	ATU Entry State. These bits are used as the Entry State for ATU Load/Purge or Flush/Move operations and it is the resulting Entry State from ATU Get Next operations (GetNext is the only ATU operation supported in the devices). If these bits equal 0x0 then the ATUOp is a Purge or a Flush. If these bits are not 0x0 then the ATUOp is a Load or a Move (a Move ATUOp requires these bits to be 0xF). On Get/Clear Violation Data ATUOps, these bits return the Source Port ID (SPID) associated with the ATU violation that was just serviced, except for Age Out violation where these return 0x1. If SPID = 0xF the source of the violation was the CPU's register interface (i.e., the ATU was full during a CPU Load operation).

- The ATU Entry State bits on Unicast ATU entries are defined as follows:
- 0x0: Unused entry
- 0x1 to 0x7: Used entry where Entry State = the Age of the entry where 0x1 is the oldest
- 0x8: Static Policy entry (Reserved on the 88E6350R/88E6350 devices)
- 0x9: Static Policy entry with Priority Override (Reserved on the 88E6350R/88E6350 devices)
- 0xA: Static Non Rate Limiting (NRL) entry or,
- 0xA: Static AVB Entry depending on MACAVB (Global 1, offset 0x0A)
- 0xB: Static Non Rate Limiting (NRL) entry with Priority Override or,
- 0xB: Static Unicast AVB Entry with Priority Override depending on MACAVB (Global 1, offset 0x0A)
- 0xC: Static entry defining frames with this DA as MGMT
- 0xD: Static entry defining frames with this DA as MGMT with Priority Override
- 0xE: Static entry
- 0xF: Static entry with Priority Override

The ATU Entry State bits on Multicast ATU entries are defined as follows:

- 0x0: Unused entry
- 0x1 to 0x3: Reserved for future use
- 0x4: Static Policy entry (Reserved on the 88E6350R/88E6350 devices)
- 0x5: Static Non Rate Limiting (NRL) entry or,
- 0x5: Static AVB Entry depending on MACAVB (Global 1, offset 0x0A)
- 0x6: Static entry defining frames with this DA as MGMT
- 0x7: Static entry
- 0x8 to 0xB: Reserved for future use
- 0xC: Static Policy entry with Priority Override (Reserved on the 88E6350R/88E6350 devices)
- 0xD: Static Non Rate Limiting (NRL) entry with Priority Override or,
- 0xD: Static AVB Entry with Priority Override depending on MACAVB (Global 1, offset 0x0A)
- 0xE: Static entry defining frames with this DA as MGMT with Priority Override
- 0xF: Static entry with Priority Override

Table 110: ATU MAC Address Register Bytes 0 & 1
Offset: 0x0D or Decimal 13

Bits	Field	Type	Description
15:8	ATUByte0	RWR	ATU MAC Address Byte 0 (bits 47:40) used as the MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. Bit 0 of byte 0 (bit 40) is the multicast bit (it is the first bit down the wire). Any MAC address with the multicast bit set to a one is considered Static by the ATU. On Get/Clear Violation Data ATUOps these bits return ATUByte0 associated with the ATU violation that was just serviced.
7:0	ATUByte1	RWR	ATU MAC Address Byte 1 (bits 39:32) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte1 associated with the ATU violation that was just serviced.

Table 111: ATU MAC Address Register Bytes 2 & 3
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
15:8	ATUByte2	RWR	ATU MAC Address Byte 2 (bits 31:24) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte2 associated with the ATU violation that was just serviced.
7:0	ATUByte3	RWR	ATU MAC Address Byte 3 (bits 23:16) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte3 associated with the ATU violation that was just serviced.

Table 112: ATU MAC Address Register Bytes 4 & 5
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
15:8	ATUByte4	RWR	ATU MAC Address Byte 4 (bits 15:8) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte4 associated with the ATU violation that was just serviced.
7:0	ATUByte5	RWR	ATU MAC Address Byte 5 (bits 7:0) used as the input MAC address for ATU Load, Purge or Get Next operations and it is the resulting MAC address from ATU Get Next operations. On Get/Clear Violation Data ATUOps, these bits return ATUByte5 associated with the ATU violation that was just serviced.



Table 113: IEEE-PRI Register
Offset: 0x18 or Decimal 24

Bits	Field	Type	Description
15:14	Tag_0x7	RWS to 0x3	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 7.
13:12	Tag_0x6	RWS to 0x3	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 6.
11:10	Tag_0x5	RWS to 0x2	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 5.
9:8	Tag_0x4	RWS to 0x2	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 4.
7:6	Tag_0x3	RWS to 0x1	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 3.
5:4	Tag_0x2	RWS to 0x1	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 2.
3:2	Tag_0x1	RWR	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 1.
1:0	Tag_0x0	RWR	IEEE 802.1p mapping. The value in this field is used as the frame's priority if its IEEE Tag has a value of 0.

Table 114: IP Mapping Table
Offset: 0x19 or Decimal 25

Bits	Field	Type	Description															
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 will be loaded into the IP Mapping register selected by the Pointer bits below (Reserved bits do not exist). After the write has taken place this bit self clears to zero.															
14	UseIPFPri	RWR	<p>Use IP Frame Priorities from this table. This bit is used to be maintain backwards compatibility.</p> <p>When this bit is cleared to a zero, the IP_FPRI data in this table is ignored. Instead the frame's initial IP_FPRI is generated by using the frame's IP_QPRI as the IP_FPRI's upper two bits, and the IP_FPRI's lowest bit comes from bit 0 of the frame's source port's Default PRI (Port offset 0x07).</p> <p>When this bit is set to a one, the IP_FPRI data in this table is used as the frame's initial IP_FPRI.</p>															
13:8	Pointer	RWR	<p>Pointer to the desired entry of the IP Mapping table. These bits select one of 64 possible IP mapping registers for both read and write operations (but not all entries exist). A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.</p> <p>When a frame is received on a port, and if its an IPv4 frame, the frame's six DiffServ bits are used to access this table to determine the frame's initial IP_QPRI and its initial IP_FPRI, depending upon the settings of the port's InitialPri and TagIfBoth bits (Port offset 0x04). If the frame is a IPv6 frame, the frame's six Traffic Class bits are used in the same way to access this table.</p> <p>The reset values in this table are as follows:</p> <table> <thead> <tr> <th>Pointer Range</th> <th>IP_QPRI</th> <th>IP_FPRI</th> </tr> </thead> <tbody> <tr> <td>0x00 to 0x0F</td> <td>0x0</td> <td>0x0</td> </tr> <tr> <td>0x10 to 0x1F</td> <td>0x1</td> <td>0x2</td> </tr> <tr> <td>0x20 to 0x2F</td> <td>0x2</td> <td>0x4</td> </tr> <tr> <td>0x30 to 0x3F</td> <td>0x3</td> <td>0x6</td> </tr> </tbody> </table>	Pointer Range	IP_QPRI	IP_FPRI	0x00 to 0x0F	0x0	0x0	0x10 to 0x1F	0x1	0x2	0x20 to 0x2F	0x2	0x4	0x30 to 0x3F	0x3	0x6
Pointer Range	IP_QPRI	IP_FPRI																
0x00 to 0x0F	0x0	0x0																
0x10 to 0x1F	0x1	0x2																
0x20 to 0x2F	0x2	0x4																
0x30 to 0x3F	0x3	0x6																
7	Reserved	RES	Reserved for future use.															
6:4	IP_FPRI	RWS see text	IPv4 and IPv6 Frame Priority Mapping. The value in this field is used as the frame's initial FPRI when the frame is an IPv4 or an IPv6 frame, and the port's InitialPri (Port offset 0x04) is configured to use IP FPRI's.															
3:2	Reserved	RES	Reserved for future use.															
1:0	IP_QPRI	RWS see text	IPv4 and IPv6 Queue Priority Mapping. The value in this field is used as the frame's initial QPRI when the frame is an IPv4 or an IPv6 frame, and the port's InitialPri and TagIfBoth registers (Port offset 0x04) are configured to use IP QPrios.															
			<p>NOTE: These bits are also accessible using the IP-PRI Mapping registers (Global 1 offsets 0x10 to 0x0F). Both access methods are supported for backwards compatibility. But new software should use this register as the function of the Global 1 offset 0x10 to 0x0F registers will be re-defined in future devices.</p>															



Link Street®

88E6321/88E6320 Functional Specification

Table 115: Monitor Control

Offset: 0x1A or Decimal 26

Bits	Field	Type	Description
15:12	Ingress Monitor Dest	RWS	<p>Ingress Monitor Destination Port. Frames that are targeted toward an Ingress Monitor Destination go out the port number indicated in these bits. This includes frames received on a DSA Tag port with the Ingress Monitor type, and frames received on a Network port that is enabled to be the Ingress Monitor Source Port (Table 70).</p> <p>If the Ingress Monitor Destination Port resides in this device these bits should point to the Network port where these frames are to egress. If the Ingress Monitor Destination Port resides in another device these bits should point to the DSA Tag port in this device that is used to get to the device that contains the Ingress Monitor Destination Port.</p>
11:8	Egress Monitor Dest	RWS	<p>Egress Monitor Destination Port. Frames that are targeted toward an Egress Monitor Destination go out of the port number indicated in these bits. This includes frames received on a DSA Tag port with the Egress Monitor type, and frames transmitted on a Network port that is enabled to be the Egress Monitor Source Port (Table 74).</p> <p>If the Egress Monitor Destination port resides in this device, these bits should point to the Network port where these frames are to egress. If the Egress Monitor Destination Port resides in another device, these bits should point to the DSA Tag port in this device that is used to reach the device that contains the Egress Monitor Destination Port.</p>

Table 115: Monitor Control (Continued)
Offset: 0x1A or Decimal 26

Bits	Field	Type	Description
7:4	CPU Dest	RWS	<p>CPU Destination Port. Many modes of frame processing need to know where the CPU is located. These modes are:</p> <p>When IGMP/MLD frame is received and Snooping is enabled on the port (port offset 0x04)</p> <p>When this port is configured as a DSA Port and it receives a To_Cpu frame¹</p> <p>When a Rsvd2CPU frames enters the port (global 2 offset 0x05)</p> <p>When the port's SA Filtering mode is Drop to CPU (port offset 0x04)</p> <p>When any of the port's Policy Options (port offset 0x0E) trap the frame to the CPU</p> <p>When the ingressing frame is a ARP and ARP mirroring is enabled in the device (port offset 0x08)</p> <p>In all cases, except for ARP, the frames that meet the enabled criteria are mapped to the port defined by this register only, overriding where the frame would normally go. In the case of ARP the frame will be mapped normally and it will also get copied to this port.</p> <p>Frames that filtered or discarded will not be mapped to the CPUDest with the exception of the Rsvd2CPU and DSA Tag cases (numbers 2 and 3).</p> <p>The CPUDest bits indicate the port number on this device where the CPU is connected (either directly or indirectly through another Marvell® switch device).</p> <p>If CPUDest = 0xF the remapped frames will be discarded, no ARP mirroring will occur and ingressing To_CPU frames will be discarded.</p> <p>NOTE: MGMT or BPDU frames detected by using the ATU are directed to the correct port where the CPU is connected by ensuring the CPU port's bit is set in the frame's MGMT DA MAC address as stored in the ATU address database (Section 3.4.5).</p>
3:0	Mirror Dest	RWS	<p>Mirror Destination Port. Frames that ingress a port that trigger a policy mirror are mapped (copied) to this port as long as the frame is not filtered or discarded. The MirrorDest should point to the port that directs these frames to the CPU that will process these frames. This target port should be a DSA Tag port so the frames will egress with a To_CPU DSA Tag with a CPU Code of Policy Mirror.</p> <p>To_CPU DSA Tag frames with a CPU Code of Policy Mirror that ingress a DSA Tag port will be sent to the port number defined in MirrorDest.</p> <p>If MirrorDest = 0xF Policy Mirroring is disabled and ingressing To_CPU Policy Mirror frames will be discarded.</p> <p>The policy mirror enable bits are configurable per port (see Policy Control, port offset 0x0E), and Miss Mirrors, Port offset 0x0D.</p>

1. To_CPU frames with a Code of Policy Mirror are mapped to the MirrorDest port (bits 3:0 of this register).



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 116: Total Free Counter

Offset: 0x1B or Decimal 27

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

Table 117: Global Control 2
Offset: 0x1C or Decimal 28

Bits	Field	Type	Description
15:14	Header Type	RWR	<p>Header Type. These bits are used to configure the bits that are placed into the Egress Header when it is enabled on a port (Port offset 0x04) as follows:</p> <ul style="list-style-type: none"> 00 = Original Header – for backwards compatibility to UniMAC's that look at Header byte 1 bits[4:2] and byte 2 bits [3:0] 01 = Single chip MGMT Header – for compatibility to Marvell Fast Ethernet switches that support Spanning Tree without DSA Tags 10 = Trunk Header – used together with the DSA Tags to perform Remote Switching 11 = Reserved for future use. <p>See section XXX on the exact content of each of these Headers.</p>
13:12	RMU Mode	RWR	<p>Remote Management Unit Mode</p> <p>0x0 = RMU feature is disabled.</p> <p>0x1 = Port 4 is enabled to be the RMU (Remote Management Unit) port for the switch.</p> <p>0x2 = Port 5 is enabled to be the RMU port for the switch.</p> <p>0x3 = Port 6 is enabled to be the RMU port for the switch.</p> <p>When RMU is enabled and this device receives a Remote Management Request frame directed to this device the frame will be processed and a Remote Management Response frame will be generated and sent out if the DA of the frame matches the conditions of the DA Check bit above. In either case, the Request frame will be discarded (as it was directed to this device).</p> <p>When RMU is disabled, Remote Management Request frames directed to this device will be discarded and ignored (i.e., it will not be processed and no Response frame will be generated).</p> <p>Regardless of the setting of these bits, Remote Management Request frames that are not directed to this device will be mapped to the port indicated by mapping the frame's Trg_Dev using the Device Mapping table (global 2, offset 0x06).</p> <p>NOTE: The setting of these bits will have no effect if the Remote Management port is in half-duplex mode. The port's FrameMode (port offset 0x05) must be DSA or EtherType DSA as well.</p> <p>The power on reset values for these bits come from RMU_MODE configuration pins.</p>
11	DA Check	RWR	<p>Check the DA on Remote Management frames. When this bit is set to a one the DA of Remote Management frames must be contained in this device's address database (ATU) as a Static entry (either unicast or multicast). If the DA of the frame is not contained in this device's address database the frame will not be processed as a Remote Management frame (i.e., it will be discarded without further action if this device is the Trg_Dev of the frame). When this bit is cleared to zero the DA of Remote Management frames is not validated before processing the frame.</p>
10:6	Reserved	RES	Reserved for future use.



Link Street®

88E6321/88E6320 Functional Specification

Table 117: Global Control 2 (Continued)

Offset: 0x1C or Decimal 28

Bits	Field	Type	Description
5	CtrMode	RWR	<p>Counter Mode. This bit controls the operating mode of the Port's Debug counter at Port offset 0x1E.</p> <p>When CtrMode is cleared to a zero the Debug counter for all ports (Port offset 0x1E) counts RxBad frames in the upper 8 bits of the register and counts RxGood frames in the lower 8 bits of the register. When this bit is set to a one the Debug counter for all ports counts Collisions in the upper 8 bits of the register and counts Tx Transmitted frames in the lower 8 bits of the register.</p> <p>The Debug Counters for all ports are cleared to a zero whenever this bit changes state (i.e., it transitions from a one to a zero or from a zero to a one).</p>
4:0	DeviceNumber	RWS to 0xXX ¹	<p>Device Number. In multi-chip systems, frames coming from a CPU (From_CPU frames) need to know when they have reached their destination chip. From_CPU frames whose Trg_Dev field matches these bits have reached their destination chip and are sent out from this chip using the port number indicated in the frame's Trg_Port field.</p> <p>The DeviceNumber value must be unique for each chip in a Multi-chip system. These bits are set at reset by the ADDR[4:0] configuration pins.</p>

1. The ADDR[4:0] configuration pins are used to set the initial value of this register. The ADDR[4:1] pins are also used to select between Multi-chip addressing mode or Single-chip addressing mode. Changing the value in this register after reset does not change the device's addressing mode nor its SMI address.

Table 118: Stats Operation Register

Offset: 0x1D or Decimal 29

Bits	Field	Type	Description
15	StatsBusy	SC	Statistics Unit Busy. This bit must be set to a one to start a Stats operation (See StatsOp below). Only one Stats operation can be executing at one time so this bit must be zero before setting it to a one. When the requested Stats operation completes, this bit automatically is cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (Table 99).
14:12	StatsOp	RWR	Statistics Unit Opcode. The devices support the following Stats operations (all of these operations can be executed while frames are transiting through the switch): <ul style="list-style-type: none"> 000 = No Operation 001 = Flush (clear) All Counters for all Ports 010 = Flush (clear) All Counters for a Port 011 = Reserved 100 = Read a Captured or Direct Counter 101 = Capture All Counters for a Port 11x = Reserved
11:10	Histogram Mode	RES to 0x3	Histogram Counters Mode. The Histogram mode bits control how the Histogram counters work as follows: <ul style="list-style-type: none"> 00 = Reserved 01 = Count received frames only 10 = Count transmitted frames only 11 = Count receive and transmitted frames
9	StatsBank	RWR	Statistics Bank of Counters. When this bit is cleared to a zero the MAC based MIBs (Bank 0) are accessed when a 'Read a Captured or Direct Counter' StatsOp is performed. When this bit is set to a one the Policy based MIBs (Bank 1) are accessed when a 'Read a Captured or Direct Counter' StatsOp is performed. <p>The value of this register has no effect on the other StatsOps commands as each StatsOp command is done to both MIB Banks at the same time.</p>
8:5	StatsPort	RWR	Access Statistics Counters directly for a Port or the Capture area. These bits can be used to directly access a ports counters without doing a capture first. Set these bits = 0x0 to access the captured counters. Set these bits = 0x1 to access the counters for Port 0. Set these bits = 0x2 to access the counters for Port 1, etc. <p>These bits represent the port number for the following StatOps:</p> <ul style="list-style-type: none"> a. Flush (clear) All Counters for a Port b. Read a Captured or Direct Counter c. Capture All Counters for a Port d. Reserved <p>These bits must be zero for all StatsOps except for Read a Captured or Direct Counter command (e.g., these bits are not used for the Flush (clear) All Counters for a Port command).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 118: Stats Operation Register (Continued)

Offset: 0x1D or Decimal 29

Bits	Field	Type	Description																																												
4:0	StatsPtr	RWR	<p>Statistics Pointer. This field is used as a parameter for the above StatsOp commands.</p> <p>StatsPtr must be set to the desired counter to read for the Read a Captured or Direct Counter (0x4) StatsOp (valid range is 0x00 to 0x1F). A Capture All Counters for a Port StatsOp must be done prior to using the Read A Captured Counter StatsOp with StatsPort = 0x0. The counter that is read is defined as follows:</p> <p>Bank 0 Ingress Counters¹ Bank 0 Egress Counters</p> <table><tr><td>0x00 – InGoodOctetsLo</td><td>0x0E – OutOctetsLo²</td></tr><tr><td>0x01 – InGoodOctetsHi</td><td>0x0F – OutOctetsHi</td></tr><tr><td>0x02 – InBadOctets</td><td></td></tr><tr><td>0x04 – InUnicast</td><td>0x10 – OutUnicast</td></tr><tr><td>0x06 – InBroadcasts</td><td>0x13 – OutBroadcasts</td></tr><tr><td>0x07 – InMulticasts</td><td>0x12 – OutMulticasts</td></tr><tr><td>0x16 – InPause</td><td>0x15 – OutPause</td></tr><tr><td>0x18 – InUndersize</td><td>0x1E – Collisions</td></tr><tr><td>0x19 – InFragments</td><td>0x05 – Deferred</td></tr><tr><td>0x1A – InOversize</td><td>0x14 – Single</td></tr><tr><td>0x1B – InJabber</td><td>0x17 – Multiple</td></tr><tr><td>0x1C – In RxErr</td><td>0x11 – Excessive</td></tr><tr><td>0x1D – InFCSErr</td><td>0x1E – Collisions</td></tr><tr><td></td><td>0x05 – Deferred</td></tr><tr><td></td><td>0x14 – Single</td></tr><tr><td></td><td>0x17 – Multiple</td></tr><tr><td></td><td>0x03 – OutFCSErr</td></tr><tr><td></td><td>0x11 – Excessive</td></tr><tr><td></td><td>0x1F – Late</td></tr></table> <p>Bank 0 Histogram Counters³</p> <table><tr><td>0x08 – 64Octets</td></tr><tr><td>0x09 – 65 to 127Octets</td></tr><tr><td>0x0A – 128 to 255Octets</td></tr><tr><td>0x0B – 256 to 511Octets</td></tr><tr><td>0x0C – 512 to 1023Octets</td></tr><tr><td>0x0D – 1024 to MaxOctets</td></tr></table>	0x00 – InGoodOctetsLo	0x0E – OutOctetsLo ²	0x01 – InGoodOctetsHi	0x0F – OutOctetsHi	0x02 – InBadOctets		0x04 – InUnicast	0x10 – OutUnicast	0x06 – InBroadcasts	0x13 – OutBroadcasts	0x07 – InMulticasts	0x12 – OutMulticasts	0x16 – InPause	0x15 – OutPause	0x18 – InUndersize	0x1E – Collisions	0x19 – InFragments	0x05 – Deferred	0x1A – InOversize	0x14 – Single	0x1B – InJabber	0x17 – Multiple	0x1C – In RxErr	0x11 – Excessive	0x1D – InFCSErr	0x1E – Collisions		0x05 – Deferred		0x14 – Single		0x17 – Multiple		0x03 – OutFCSErr		0x11 – Excessive		0x1F – Late	0x08 – 64Octets	0x09 – 65 to 127Octets	0x0A – 128 to 255Octets	0x0B – 256 to 511Octets	0x0C – 512 to 1023Octets	0x0D – 1024 to MaxOctets
0x00 – InGoodOctetsLo	0x0E – OutOctetsLo ²																																														
0x01 – InGoodOctetsHi	0x0F – OutOctetsHi																																														
0x02 – InBadOctets																																															
0x04 – InUnicast	0x10 – OutUnicast																																														
0x06 – InBroadcasts	0x13 – OutBroadcasts																																														
0x07 – InMulticasts	0x12 – OutMulticasts																																														
0x16 – InPause	0x15 – OutPause																																														
0x18 – InUndersize	0x1E – Collisions																																														
0x19 – InFragments	0x05 – Deferred																																														
0x1A – InOversize	0x14 – Single																																														
0x1B – InJabber	0x17 – Multiple																																														
0x1C – In RxErr	0x11 – Excessive																																														
0x1D – InFCSErr	0x1E – Collisions																																														
	0x05 – Deferred																																														
	0x14 – Single																																														
	0x17 – Multiple																																														
	0x03 – OutFCSErr																																														
	0x11 – Excessive																																														
	0x1F – Late																																														
0x08 – 64Octets																																															
0x09 – 65 to 127Octets																																															
0x0A – 128 to 255Octets																																															
0x0B – 256 to 511Octets																																															
0x0C – 512 to 1023Octets																																															
0x0D – 1024 to MaxOctets																																															

Table 118: Stats Operation Register (Continued)
Offset: 0x1D or Decimal 29

Bits	Field	Type	Description
4:0 (cont.)	StatsPtr (cont.)	RWR (cont.)	Bank 1 Ingress Counters 0x00 – InDiscards 0x01 – InFiltered 0x02 – InAccepted 0x03 – InBadAccepted 0x04 – InGoodAvbClassA 0x05 – InGoodAvbClassB 0x06 – InBadAvbClassA 0x07 – InBadAvbClassB Bank 1 Egress Counters 0x10 – OutQueue0 0x11 – OutQueue1 0x12 – OutQueue2 0x13 – OutQueue3 0x14 – (OutQueue4) 0x15 – (OutQueue5) 0x16 – (OutQueue6) 0x17 – (OutQueue7) 0x08 – TCAMCounter0 0x09 – TCAMCounter1 0x0A – TCAMCounter2 0x0B – TCAMCounter3 0x0C – 0x0D – 0x0E – InDaUnknown 0x0F – InMGMT 0x18 – OutCutThrough 0x19 – 0x1A – OutOctetsA 0x1B – OutOctetsB 0x1C – 0x1D – 0x1E – 0x1F – OutMGMT

1. If Marvell® Header mode is used on Ports 0 to 4 the extra two bytes in the frame are not included in the InGoodOctet nor the InBadOctet counts.
2. OutOctets may not accurately count the bytes transmitted on frames that encounter a collision.
3. If Marvell Header mode is used on Ports 0 to 4 the extra two bytes in the frame are not included in the count before determining which Histogram Counter to increment.



Link Street®

88E6321/88E6320 Functional Specification

Table 119: Stats Counter Register Bytes 3 & 2
Offset: 0x1E or Decimal 30

Bits	Field	Type	Description
15:8	StatsByte3	RO	Statistics Counter Byte 3. These bits contain bits 31:24 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 118). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.
7:0	StatsByte2	RO	Statistics Counter Byte 2. These bits contain bits 23:16 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 118). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.

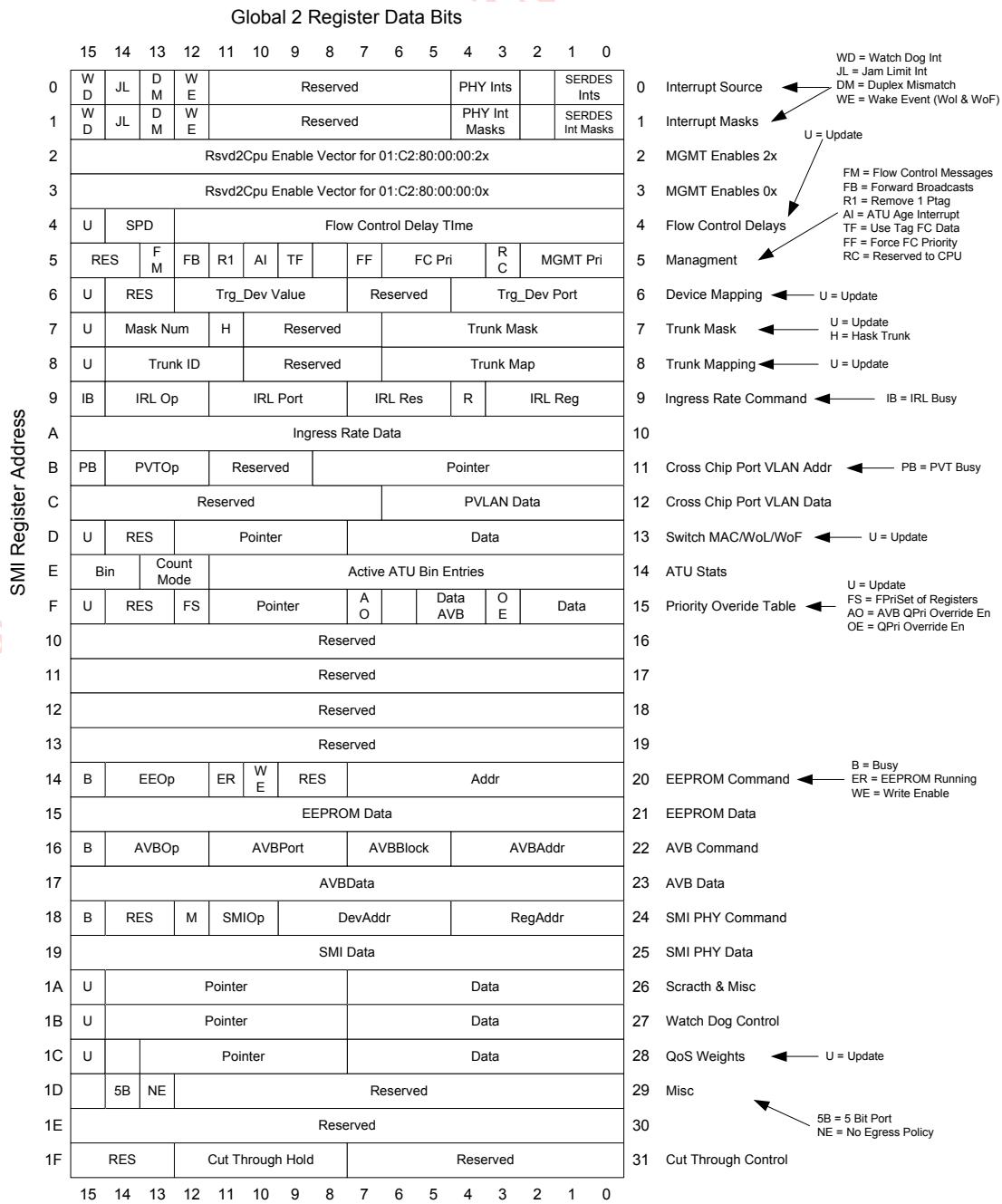
Table 120: Stats Counter Register Bytes 1 & 0
Offset: 0x1F or Decimal 31

Bits	Field	Type	Description
15:8	StatsByte1	RO	Statistics Counter Byte 1. These bits contain bits 15:8 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 118). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.
7:0	StatsByte0	RO	Statistics Counter Byte 0. These bits contain bits 7:0 of the last stat counter requested to be read by the CPU (by using the Read a Counter StatsOp — Table 118). They will return data from the MAC based MIBs (Bank 0) when StatsBank (Global 1 offset 0x1D) is cleared to a zero, and they will return data from the Policy based MIBs (Bank 1) when StatsBank is set to a one.

10.4.2 Switch Global 2 Registers

The devices contain a second set of global registers that effect all the Ethernet ports in the device. Each Global 2 register is 16-bits wide and their bit assignment are shown in Figure 65.

Figure 65: Global 2 Register bit Map (Device Addr 0x1C)





Link Street®

88E6321/88E6320 Functional Specification

Table 121: Interrupt Source Register

Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	WatchDog Int	RO	WatchDog interrupt. This bit indicates a watch dog event occurred. Watch Dog events are enabled in the Watch Dog Control register (Global 2, offset 0x1B).
14	JamLimit	ROC	Jam Limit interrupt. This bit is set to a one when any of the ports detect an Ingress Jam Limit violation as determined by the port's LimitIn setting (in Jamming Control, port offset 0x02).
13	Duplex Mismatch	ROC	Duplex Mismatch interrupt. This bit is set to a one when any of the ports detect a duplex mismatch (i.e., the local port is in half duplex mode while the link partner is in full duplex mode). When this bit is set to a one, the port that the duplex mismatch was detected on will be seen in the DMPort register (Global 2 offset 0x1B, index 0x40). Software must set the DMPort register back to a value of 0xF to re-arm the Duplex Mismatch interrupt.
12	WakeEvent	RO	Wake Event interrupt. This bit is set to a one when any of the ports detect an enabled Wake Event such as Wake on LAN or Wake on Frame. When this bit is a one, software can find the source of the interrupt by reading the WoF Int bit and the WoL Port bits (both registers are in index 0x1F of Switch MAC/WoL/WoF at Global 2 offset 0x0D).
11:5	Reserved	RES	Reserved for future use.
4:3	PHYInt	RO	PHY layer core interrupt bit. This bit is set when any of the internal PHY core's interrupt bit is set. A port's PHYInt bit will clear to zero when all the unmasked interrupts from the port's PHY are serviced. Bit 4 is for port 4, bit 3 is for port 3 etc.
2	Reserved	RES	Reserved for future use.
1:0	SERDES Int	RO	SERDES layer core interrupt bit. This bit is set when any of the internal SERDES core's interrupt bits are set. A port's SERDESInt bit will clear to zero when all the unmasked interrupts from the port's SERDES are serviced. Bit 1 is for port 1, bit 0 is for port 0 etc.

Table 122: Interrupt Mask Register
Offset: 0x01 or Decimal 1

Bits	Field	Type	Description
15	WatchDog IntEn	RWR	WatchDog interrupt enable. This bit must be set to a one to allow the WatchDog interrupt (global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (global offset 0x00) so that the INTn pin can be driven low.
14	JamLimitEn	ROC	Jam Limit interrupt enable. This bit must be set to a one to allow the JamLimit interrupt (global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (global offset 0x00) so that the INTn pin can be driven low.
13	Duplex Mismatch Error	RWR	Duplex Mismatch interrupt enable. This bit must be set to a one to allow the Duplex Mismatch interrupt (Global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (Global offset 0x00) so that the INTn pin can be driven low.
12	WakeEventEn	RWR	Wake Event interrupt enable. This bit must be set to a one to allow the WakeEvent interrupt (Global 2 offset 0x00) to drive the DeviceInt bit in the Switch Global Status register (Global offset 0x00) so that the INTn pin can be driven low.
11:5	Reserved	RES	Reserved for future use.
4:3	PHYIntEn	RWR	PHY layer core interrupt enable bit. This bit is set to a one to allow PHYinterrupts from a given physical layer core to drive the DeviceInt bit in the Switch Global Status register (Global Offset 0x00) so that the INTn pin can be driven low. Bit 4 is for Port 4, bit 3 is for Port 3 etc.
2	Reserved	RES	Reserved for future use.
1:0	SERDES IntEn	RWR	SERDES layer core interrupt enable bit. This bit is set to a one to allow SERDES Interrupts from a given physical layer core to drive the DeviceInt bit in the Switch Global Status register (Global offset 0x00) so that the INTn pin can be driven low. Bit 1 is for port 1, bit 0 is for port 0 etc.



Link Street®

88E6321/88E6320 Functional Specification

Table 123: MGMT Enable Register 2x

Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15:0	Rsvd2CPU Enables 2x	RWS	<p>Reserved DA Enables 2x. When the Rsvd2Cpu bit (Global 2, offset 0x05) is set to a one the 16 reserved multicast DA addresses whose bit in this register are also set to a one, are treated as MGMT¹ frames. The reserved DA's supported by this register take the form 01:80:C2:00:00:2x. When x = 0x0, bit 0 of this register is tested. When x = 0x2 bit 2 of this field is tested and so on with x = 0xF bit 15 of this register is tested.</p> <p>If the tested bit in this register is cleared to a zero, the frame will be treated as a normal (non-MGMT) frame.</p> <p>This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames.</p> <p>If the Rsvd2Cpu bit (Global 2, offset 0x05) is cleared to a zero these bits will have no effect.</p>

1. MGMT, or management, frames are used for managed switch protocols like GVRP. The switch processes MGMT frames differently.

Table 124: MGMT Enable Register 0x

Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:0	Rsvd2CPU Enables 0x	RWS	<p>Reserved DA Enables 0x. When the Rsvd2Cpu bit (Global 2, offset 0x05) is set to a one the 16 reserved multicast DA addresses whose bit in this register are also set to a one, are treated as MGMT¹ frames. All the reserved DA's supported by this register take the form 01:80:C2:00:00:0x. When x = 0x0, bit 0 of this register is tested. When x = 0x2 bit 2 of this field is tested and so on with x = 0xF bit 15 of this register is tested.</p> <p>If the tested bit in this register is cleared to a zero, the frame will be treated as a normal (non-MGMT) frame.</p> <p>This register allows some or all of these 16 reserved multicast addresses to be treated as MGMT frames².</p> <p>If the Rsvd2Cpu bit (Global 2, offset 0x05) is cleared to a zero these bits will have no effect.</p>

1. MGMT, or management, frames are used for managed switch protocols link Spanning Tree (STP) and Link Aggregation (LAC). The switch processes MGMT frames differently (see [Section 5.9](#)).
2. Frames with a DA of 01:80:C2:00:00:01 (the Pause frame DA) are always treated as MAC control frames and cannot be treated as MGMT frames.

Table 125: Flow Control Delay Register
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15	Update	SC	Update FC Delay Time data. When this bit is set to a one the data written to bits 12:0 will be loaded into the FC Delay Time register selected by the SPD bits below. After the write has taken place this bit self clears to zero.
14:13	SPD	RWR	Speed Number. These bits select one of three possible FC Delay Time register for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
12:0	FC Delay Time	RWS	<p>Flow Control Delay Time. These bits are used to cause a MAC to assert Flow Control for the delay amount times 8.192 uSecs. The register used is determined by the Flow Control DSA Tag frame's SPD bits that was directed at this MAC.</p> <p>Three FC Delay Time registers are accessed by using the SPD bits above. SPD 0b00 is assigned for as the Flow Control delay to use when talking to 10 Mbit ports. SPD 0b01 is assigned for 100 Mbit ports and SPD 0b10 is assigned for 1000 Mbit ports (SPD of 0b11 is reserved for future use and should not be accessed). The default values for each of these registers are shown below.</p> <p>SPD 0b00 resets to 0x0258 (600 decimal) SPD 0b01 resets to 0x003C (60 decimal) SPD 0b10 resets to 0x0006 (6 decimal)</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 126: Switch Management Register
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	Loopback Filter	RWR	Loopback filter. When this bit is cleared to a zero, normal operation occurs. When this bit is set to a one, Forward DSA frames that Ingress a DSA port that came from the same Src_Dev will be filtered to the same Src_Port (i.e., the frame will not be allowed to egress the source port on the source device as indicated in the DSA Forward's Tag).
14	Reserved	RES	Reserved for future use.
13	Flow Control Message	RWR or RWS ¹	Enable Flow Control Messages ² . When this bit is set to a one DSA Tag Flow Control messages will be generated when a Flow Control enabled output queue becomes congested. When this bit is cleared to a zero DSA Tag Flow Control messages will not be generated but any received will be processed at the target MAC if flow control is enabled on the target MAC.
12	FloodBC	RWR	Flood Broadcast. When this bit is set to a one frames with the Broadcast destination address will flood out all the ports regardless of the setting of the port's Egress Floods bits (in Port Control, offset 0x04). VLAN rules and other switch policy still applies to these Broadcast frames. This bit only changes the policy of the Default Forward bit for Broadcast frames. When this bit is cleared to a zero frames with the Broadcast destination address are considered Multicast frames and will not egress out ports that have their Egress Flood bit cleared unless the Broadcast address is found in the address database.
11	Remove 1PTag	RWR	Remove One Provider Tag. When this bit is set to a one and a port is configured as a Provider Port (EgressMode = 0x3 in Port Control, port offset 0x04), recursive Provider Tag stripping will NOT be performed. Only the first Provider Tag found on the frame will be extracted and removed. Its extracted data will be used for switching. When this bit is cleared to a zero and a port is configured as a Provider Port (EgressMode = 0x3 in Port Control, port offset 0x04), recursive Provider Tag stripping will be performed. The first Provider Tag's data will be extracted and used for switching, and then all subsequent Provider Tags found in the frame will also be removed. This will only occur if the port's PortEType register (used to define the Provider Tag's EtherType) is not 0x8100 (can't perform recursive Provider Tag removal when the Provider's EtherType is equal to 0x8100).
10	ATUAge IntEn	RWS	ATU Age Violation Interrupt Enable. When a port is Locked (port offset 0x0B) an ATU Miss Violation will be generated when the frame's SA is not found in the address database. When this bit is set to a one an ATU Miss Violation will also be generated when a frame's SA is found in the address database but it has an Entry State value less than 0x4 (i.e., it is about half way aged out). RefreshLocked (port offset 0x0B) must not be enabled for this Age Violation to occur. Adding the ATU Age Violation to the ATU Miss Violation allows CPU directed learning to know an address is still being used before it ages out.

Table 126: Switch Management Register (Continued)
Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
9	Tag Flow Control	RWR	Use and generate source port Flow Control status for Cross-Chip Flow Control. When this bit is set to a one bit 17 of the DSA Tag Forward frames is defined to be Src_FC and it is added to these frames when generated and it is inspected on these frames when received. When this bit is clearer to a zero bit 17 of the DSA Tag Forward frames is defined to be Reserved and it will be zero on these frames when generated and it will not be used on these frames when received (this is a backwards compatibility mode).
8	Reserved	RES	Reserved for future use.
7	ForceFlow ControlPri	RWS	Force Flow Control Priority. When this bit is set to a one the PRI[2:0] bits of generated DSAI Tag Flow Control frames will be set to the value of the FC Pri bits below. When this bit is cleared to a zero generated DSA Tag Flow Control frames will retain the PRI[2:0] bits from the frame that caused the congestion. This bit will have no effect if the FlowControlMessage bit (above) is cleared to a zero.
6:4	FC Pri	RWS to 0x7	Flow Control Priority. These bits are used as the PRI[2:0] bits on generated DSA Tag Flow Control frames if the ForceFlowControlPri bit above is set to a one.
3	Rsvd2CPU		<p>Reserved multicast frames to CPU. This device supports two ways to support protocols that use multicast addresses.</p> <p>The first way is to enter the multicast address into the address database with a MGMT Entry_State, mapping it toward the CPU's port (Table 108). This allows proprietary protocols to be supported while also supporting standard protocols. If multiple address databases are used each multicast address will need to be added to the database for each database.</p> <p>The second way is to set this bit to a one. When this bit is a one frames with a Destination Address in the range 01:80:C2:00:00:0x or 01:80:C2:00:00:2x, regardless of their VLAN membership, will be considered MGMT frames and sent to the port's CPUDestPort (global offset 0x1A) as long as the associated Rsvd2Cpu Enable bit for the frame's DA is also set to a one (Global 2 offset 0x02 and 0x03). The MGMT Pri field (below) is used as the priority on these frames.</p>
2:0	MGMT Pri	RWS to 0x7	MGMT Priority. These bits are used as the priority to use on Rsvd2CPU frames (above).

1. The FlowControlMessage bit will set to a one (enabled) if the HD_FLOW configuration pin is high and the FD_FLOW configuration pin is low at the end of the configuration time following the rising edge of RESETn. This combination of configuration pins enables Cross-chip Flow Control on all Network ports when these ports are in either full or half-duplex mode of operation.
2. Flow Control Messages will egress out DSA links only, when the frame received on this link caused congestion. When Flow Control Messages are used the DSA link must have Flow Control enabled or some frame loss will occur.



Link Street®

88E6321/88E6320 Functional Specification

Table 127: Device Mapping Table Register
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description																																																																				
15	Update	SC	Update Target Device Routing data. When this bit is set to a one the data written to bits 3:0 will be loaded into the Target Device entry selected by the Trg_DevValue bits below. After the write has taken place this bit self clears to zero.																																																																				
14:13	Reserved	RES	Reserved for future use.																																																																				
12:8	Trg_Dev Value	RWR	Target Device Value. These bits select one of 32 possible Target Device Port register for both read and write operations to the Mapping Table. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.																																																																				
7:4	Reserved	RES	Reserved for future use.																																																																				
3:0	Trg_Dev Port	RWS	<p>Target Device Port number. These bits point to the physical port on this device where From_CPU frames will be routed by using the frame's Trg_Dev as an index into this table (when the Cascade Port, Global Control 2, Offset 0x1C, is set to a value of 0xF). In this way a physical mapping, or Routing Table, of the interconnection of the devices that make up the switch box or boxes in a stack is defined.</p> <p>When a write occurs to this register with the Update bit being a one these bits are written to the Trg_Dev Port selected by the Trg_Dev Value bits. When a write occurs to this register with the Update bit being a zero these bits are not written anywhere (this allow the Trg_Dev Value bits to be written to for read operations). When a read occurs to this register these bits reflect the Target Device Port data found for the entry selected by the Trg_Dev Value bits.</p> <p>The Routing Table is reset to the following values:</p> <table><thead><tr><th>Trg_Dev Value</th><th>Trg_Dev Port</th><th>Trg_Dev Value</th><th>Trg_Dev Port</th></tr></thead><tbody><tr><td>0x00</td><td>0x0</td><td>0x10</td><td>0x0</td></tr><tr><td>0x01</td><td>0x1</td><td>0x11</td><td>0x1</td></tr><tr><td>0x02</td><td>0x2</td><td>0x12</td><td>0x2</td></tr><tr><td>0x03</td><td>0x3</td><td>0x13</td><td>0x3</td></tr><tr><td>0x04</td><td>0x4</td><td>0x14</td><td>0x4</td></tr><tr><td>0x05</td><td>0x5</td><td>0x15</td><td>0x5</td></tr><tr><td>0x06</td><td>0xF</td><td>0x16</td><td>0xF</td></tr><tr><td>0x07</td><td>0xF</td><td>0x17</td><td>0xF</td></tr><tr><td>0x08</td><td>0xF</td><td>0x18</td><td>0xF</td></tr><tr><td>0x09</td><td>0xF</td><td>0x19</td><td>0xF</td></tr><tr><td>0x0A</td><td>0xF</td><td>0x1A</td><td>0xF</td></tr><tr><td>0x0B</td><td>0xF</td><td>0x1B</td><td>0xF</td></tr><tr><td>0x0C</td><td>0xF</td><td>0x1C</td><td>0xF</td></tr><tr><td>0x0D</td><td>0xF</td><td>0x1D</td><td>0xF</td></tr><tr><td>0x0E</td><td>0xF</td><td>0x1E</td><td>0xF</td></tr><tr><td>0x0F</td><td>0xF</td><td>0x1F</td><td>0xF</td></tr></tbody></table>	Trg_Dev Value	Trg_Dev Port	Trg_Dev Value	Trg_Dev Port	0x00	0x0	0x10	0x0	0x01	0x1	0x11	0x1	0x02	0x2	0x12	0x2	0x03	0x3	0x13	0x3	0x04	0x4	0x14	0x4	0x05	0x5	0x15	0x5	0x06	0xF	0x16	0xF	0x07	0xF	0x17	0xF	0x08	0xF	0x18	0xF	0x09	0xF	0x19	0xF	0x0A	0xF	0x1A	0xF	0x0B	0xF	0x1B	0xF	0x0C	0xF	0x1C	0xF	0x0D	0xF	0x1D	0xF	0x0E	0xF	0x1E	0xF	0x0F	0xF	0x1F	0xF
Trg_Dev Value	Trg_Dev Port	Trg_Dev Value	Trg_Dev Port																																																																				
0x00	0x0	0x10	0x0																																																																				
0x01	0x1	0x11	0x1																																																																				
0x02	0x2	0x12	0x2																																																																				
0x03	0x3	0x13	0x3																																																																				
0x04	0x4	0x14	0x4																																																																				
0x05	0x5	0x15	0x5																																																																				
0x06	0xF	0x16	0xF																																																																				
0x07	0xF	0x17	0xF																																																																				
0x08	0xF	0x18	0xF																																																																				
0x09	0xF	0x19	0xF																																																																				
0x0A	0xF	0x1A	0xF																																																																				
0x0B	0xF	0x1B	0xF																																																																				
0x0C	0xF	0x1C	0xF																																																																				
0x0D	0xF	0x1D	0xF																																																																				
0x0E	0xF	0x1E	0xF																																																																				
0x0F	0xF	0x1F	0xF																																																																				

Table 128: Trunk Mask Table Register

Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15	Update	SC	Update Trunk Mask data. When this bit is set to a one the data written to bits 10:0 will be loaded into the Trunk Mask selected by the MaskNum bits below. After the write has taken place this bit self clears to zero.
14:12	MaskNum	RWR	Mask Number. These bits select one of eight possible Trunk Mask vectors for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
11	HashTrunk	RWR	Hash DA & SA for TrunkMask selection. Trunk load balancing is accomplished by using the frame's DA and SA fields to access one of eight Trunk Masks, unless the TCAM's Load Balance Override is set. When this bit is set to a one the hash computed for DA and SA address table lookups is used for the TrunkMask selection. When this bit is cleared to a zero the lower 3 bits of the frame's DA and SA are XOR'ed together to select the TrunkMask to use. When the TCAM's Load Balance Override is set on a frame, this bit has no effect on that frame as the TCAM's Load Balance Data is used instead.
10:7	Reserved	RES	Reserved for future use.
6:0	TrunkMask	RWS	Trunk Mask bits. Bit 0 controls trunk masking for port 0, bit 1 for port 1, etc. When a write occurs to this register with the Update bit being a one these bits are written to the Trunk Mask selected by the MaskNum bits. When a write occurs to this register with the Update bit being a zero these bits are not written anywhere (this allow the MaskNum bits to be written to for read operations). When a read occurs to this register these bits reflect the Trunk Mask data found for the entry selected by the MaskNum bits. The TrunkMask is reset to all ones for all MaskNum entries.

Table 129: Trunk Mapping Table Register

Offset: 0x08 or Decimal 8

Bits	Field	Type	Description
15	Update	SC	Update Trunk Routing data. When this bit is set to a one the data written to bits 10:0 will be loaded into the Trunk Route selected by the Trunk ID bits below. After the write has taken place this bit self clears to zero.
14:11	Trunk ID	RWR	Trunk Identifier. These bits select one of sixteen possible Trunk ID routing vectors for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
10:7	Reserved	RES	Reserved for future use.
6:0	TrunkMap	RWR	Trunk Map bits. Bit 0 controls trunk routing for port 0, bit 1 for port 1, etc. When a write occurs to this register with the Update bit being a one these bits are written to the Trunk Map selected by the Trunk ID bits. When a write occurs to this register with the Update bit being a zero these bits are not written anywhere (this allow the Trunk ID bits to be written to for read operations). When a read occurs to this register these bits reflect the Trunk Mapping data found for the entry selected by the Trunk ID bits.



Link Street®

88E6321/88E6320 Functional Specification

Table 130: Ingress Rate Command Register
Offset: 0x09 or Decimal 9

Bits	Field	Type	Description
15	IRLBusy	SC	Ingress Rate Limit unit Busy. This bit must be set to a one to start an IRL operation (see IRLOp below). Only one IRL operation can be executing at one time so this bit must be zero before setting it to a one. When the requested IRL operation completes this bit will automatically be cleared to a zero.
14:12	IRLOp	RWR	Ingress Rate Limit unit Opcode. The devices support the following IRL operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Init all resources to the initial state 010 = Init the selected resource (pointed to by IRLPort and IRLRes) to the initial state. This initializes internal rate limiting related counters. 011 = Write to the selected resource/register (IRLUnit/IRLReg) 100 = Read the selected resource/register (IRLUnit/IRLReg) 101 = Reserved 110 = Reserved 111 = Reserved
11:8	IRLPort	RWR	Ingress rate limiting port. These bits indicate the ingress rate limiting port that is being accessed. Since there are 11 ports in the devices, these bits indicate one of the eleven ports. For example, if this field is programmed to a 0x3, it indicates that ingress rate resource belonging to port number 3 is being accessed.
7:5	IRLRes	RWR	Ingress rate limit resource. These bits indicate the ingress rate limit resource number that is being accessed. Since there are five rate limiting resources per port, these bits indicate one of the five resources. For example, if this field is programmed to 0x2, it indicates that ingress rate resource 2 is being accessed.
4	Reserved	RWR	Reserved for future use.
3:0	IRLReg	RWR	Ingress Rate Limit register. These bits are used to define the controlling register being written or read on the resource defined in IRLUnit above. Use a value of 0x0 to access register 0, a value of 0x1 to access register 1, etc.

Table 131: Ingress Rate Data Register
Offset: 0x0A or Decimal 10

Bits	Field	Type	Description
15:0	IRLData	RWR	<p>Ingress Rate Limit Data. These data bits are either the read data or the write data bits depending on the PIRL Command register (Global 2 offset 0x09).</p> <p>In the case of a read operation, the hardware logic fetches the data bits from the specified address in the PIRL Command register and stores them into these bits. In the case of a write operation, the hardware logic utilizes the data bits in this field to write to the specified address location in the PIRL Command register.</p> <p>The content of the PIRL registers is documented in Section 1.3.4 on page 126.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 132: Cross-chip Port VLAN Register

Offset: 0x0B or Decimal 11

Bits	Field	Type	Description
15	PVTBusy	SC	Port VLAN Table Busy. This bit must be set to a one to start a PVT operation (see PVTOp below). Only one PVT operation can be executing at one time so this bit must be zero before setting it to a one. When the requested PVT operation completes this bit will automatically be cleared to a zero.
14:12	PVTOp	RWR	Port VLAN Table Opcode. The device supports the following PVT operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Init the PVT Table to all one's (initial state) 010 = Reserved 011 = Write PVLAN Data (global 2, offset 0x0C) to the selected register ¹ 100 = Read the selected register ² 101 = Reserved 110 = Reserved 111 = Reserved
11:9	Reserved	RES	Reserved for future use.
8:0	Pointer	RWR	Pointer to the desired entry of the Cross-chip Port VLAN Table. These bits select one of 512 possible table entries for both read and write operations (defined by the PVTOp bits above). The meaning of the data bits in the table is described in the Cross-chip Port VLAN Data register below (global 2 offset 0x0C).

1. The register that gets written is the one pointed to by the Pointer register bits (bit 8:0 of this register)

2. The register that gets read is the one pointed to by the Pointer register bits (bit 8:0 of this register)

Table 133: Cross-chip Port VLAN Data Register
Offset: 0x0C or Decimal 12

Bits	Field	Type	Description
15:7	Reserved	RES	Reserved for future use.
6:0	PVLAN Data	RWS	<p>Cross-chip Port VLAN Data used as a bit mask to limit where Cross-chip frames can egress (In-chip Port VLANs are masked using the VLANTable - port offset 0x06). Cross-chip frames are Forward frames that ingress a DSA or Ether Type DSA port (see Frame Mode in port offset 0x04)¹. Bit 0 is a mask for port 0, bit 1 for port 1, etc. When a port's mask bit is one frames are allowed to egress that port on this device. When a port's mask bit is zero frames are not allowed to egress that port on this device.</p> <p>The entries in the Cross-chip Port VLAN Table are read and loaded by a CPU with the Cross-chip Port VLAN Addr register above (global 2 offset 0x0B).</p> <p>The 512 entry Cross-chip Port VLAN Table is accessed by ingressing frames based upon the original source port of the frame using the Forward frame's DSA tag fields Src_Dev, Src_Port/Src_Trunk and Src_Is_Trunk. The entry that is accessed by the frame is:</p> <ul style="list-style-type: none"> If 5 Bit Port (in Global 2, offset 0x1D) = 0: If Src_Is_Trunk = 0 -> Src_Dev[4:0], Src_Port[3:0]² If Src_Is_Trunk = 1 -> 0x1F, Src_Trunk[3:0] (i.e., at Src_Dev 0x1F) If 5 Bit Port (in Global 2, offset 0x1D) = 1: If Src_Is_Trunk = 0 -> Src_Dev[3:0], Src_Port[4:0]³ If Src_Is_Trunk = 1 -> 0xF, Src_Trunk[4:0] (i.e., at Src_Dev 0x0F) <p>Cross-chip port VLANs with Trunks are supported in the table where this device's entries would be stored (defined by this device's Device Number). This portion of the table is available for Trunk entries because this device's port VLAN mappings to ports inside this device are masked by the port's VLANTable (port offset 0x06).</p>

1. Cross-chip port VLANs cannot be supported on Ether Type DSA ports on Forward frames that don't contain a DSA Tag (Non-Forward DSA frames are not filtered by this table).
2. Only the lower 4 bits of the Src_Port are needed when interconnecting 88E6xxx switch devices since they all support less than 16 physical ports.
3. The full 5 bits of the Src_Port are needed when interconnecting this device with 98DXxxx switch devices since they support more than 16 physical ports. Only 16 Devices are supported in this mode, however.



Link Street®

88E6321/88E6320 Functional Specification

Table 134: Switch MAC/WoL/WoF Register
Offset: 0x0D or Decimal 13

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

Table 135: Switch MAC Byte 0, Register Index: 0x00 of Switch MAC/WoL/WoF

Bits	Field	Type	Description
7:1	SwMAC Byte 0	RWR	Switch MAC Address Byte 0 (bits 47:41). Used as the switch's source address (SA) in transmitted full-duplex Pause frames. Since bit 0 of byte 0 (bit 40) is the multicast bit (it is the 1st bit down the wire) it is always transmitted as a zero and its value cannot be changed.
0	DiffAddr	RWR	Different MAC addresses per Port. This bit is used to have all ports transmit the same or different source addresses in full-duplex Pause frames. When this bit = 0, all ports transmit the same SA. When this bit = 1, each port uses a different SA where bit 47:4 of the MAC address are the same, but bit 3:0 are the port number (Port 0 = 0, Port 1 = 1, etc.).

Table 136: Switch MAC Byte 1, Register Index: 0x01 of Switch MAC/WoL/WoF

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

Table 137: Switch MAC Byte 2, Register Index: 0x02 of Switch MAC/WoL/WoF

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

Table 138: Switch MAC Byte 3, Register Index: 0x03 of Switch MAC/WoL/WoF

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

Table 139: Switch MAC Byte 4, Register Index: 0x04 of Switch MAC/WoL/WoF

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



Link Street®

88E6321/88E6320 Functional Specification

Table 140: Switch MAC Byte 5, Register Index: 0x05 of Switch MAC/WoL/WoF

Bits	Field	Type	Description
7:0	SwMAC Byte 5	RWR	Switch MAC Address Byte 4 (bit 7:0). Used as the switch's source address (SA) in transmitted full-duplex Pause frames.

Table 141: Wake on Frame Int Source, Register Index: 0x0C of Switch MAC/WoL/WoF

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

Table 142: Wake on Frame Int Enable, Register Index: 0x0E of Switch MAC/WoL/WoF

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

Table 143: WoL MAC Byte 0, Register Index: 0x10 of Switch MAC/WoL/WoF

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

Table 144: WoL MAC Byte 1, Register Index: 0x11 of Switch MAC/WoL/WoF

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



Link Street®

88E6321/88E6320 Functional Specification

Table 145: WoL MAC Byte 2, Register Index: 0x12 of Switch MAC/WoL/WoF

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

Table 146: WoL MAC Byte 3, Register Index: 0x13 of Switch MAC/WoL/WoF

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

Table 147: WoL MAC Byte 4, Register Index: 0x14 of Switch MAC/WoL/WoF

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

Table 148: WoL MAC Byte 5, Register Index: 0x15 of Switch MAC/WoL/WoF

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

Table 149: WoL Password Byte 0, Register Index: 0x16 of Switch MAC/WoL/WoF

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

Table 150: WoL Password Byte 1, Register Index: 0x17 of Switch MAC/WoL/WoF

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

Table 151: WoL Password Byte 2, Register Index: 0x18 of Switch MAC/WoL/WoF

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

Table 152: WoL Password Byte 3, Register Index: 0x19 of Switch MAC/WoL/WoF

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

Table 153: WoL Password Byte 4, Register Index: 0x1A of Switch MAC/WoL/WoF

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

Table 154: WoL Password Byte 5, Register Index: 0x1B of Switch MAC/WoL/WoF

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



Link Street®

88E6321/88E6320 Functional Specification

Table 155: WoL Control Byte 0, Register Index: 0x1C of Switch MAC/WoL/WoF

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

Table 156: WoL Control Byte 1, Register Index: 0x1D of Switch MAC/WoL/WoF

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

Table 157: WoL Control Byte 2, Register Index: 0x1E of Switch MAC/WoL/WoF

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

Table 158: Wake Interrupt Source, Register Index: 0x1F of Switch MAC/WoL/WoF

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



Link Street®

88E6321/88E6320 Functional Specification

Table 159: ATU Stats Register
Offset: 0x0E or Decimal 14

Bits	Field	Type	Description
15:14	Bin	RWR	Bin selector bits. These bits are used to access the 4 Bin counters for static or non-static entries readable in bits 10:0 below. A value of 0x0 will access the lowest, or 1st bin to fill counter. 0x1 will access bin 1 and 0x2 will access bin 2's counter. A value of 0x3 will access the lowest, or last to fill bin counter.
13:12	CountMode	RWR	Bin Counter Mode. These bits determine what ATU entries are counted in the four Bin counters so various information can be extracted as follows: 00 = Count all valid entries 01 = Count all valid non-static entries only 10 = Count all valid entries found in the defined FID only 11 = Count all valid non-static entries found in the defined FID only The defined FID is the FID used during the ATU GetNext operation. These bits must be set prior to the start of an ATU GetNext so the ActiveBinCtrs contain this selected data at the end of the ATU GetNext.
11:0	ActiveBin Ctr	RO	Active ATU Bin Entry Counter. When a ATU GetNext operation is started the four Bin counters are all cleared to zero. When the ATU GetNext completes these four counters can be read and added together to get a total number of active MAC addresses that were currently found in the address data base using the CountMode above. Bin 0 is 1st bin to be used. Bin 1 is used when a Hash collision occurs and Bin 0 is already used. Bin 2 is used only after both bin 0 and 1 are filled, etc.

Table 160: Priority Override Table
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 3:0 will be loaded into the QPri_AVB or FPri Override register selected by the FPriSet and Pointer bits below. After the write has taken place this bit self clears to zero.
14:13	Reserved	RES	Reserved for future use.
12	FPriSet	RWR	<p>When this bit is cleared to a zero the reading and writing actions of bits 7:0 below access the QPri and QPriAVB entries of the Priority Override table for the frame type determined by the Pointer bits below.</p> <p>When this bit is set to a one, the reading and writing actions of bits 7:0 below access the FPri entries of the Priority Override table for the frame type determined by the Pointer bits below.</p> <p>NOTE: This Priority Override table is accessed one set at a time (QPri or FPri) by this register interface.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 160: Priority Override Table
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
11:8	Pointer	RWT	<p>Pointer to the desired entry of the Priority Override table. These bits select one of sixteen possible QPri, Qpri_AVB, or FPri Override registers for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs. Each entry in the table can be used on the following frame types:</p> <p>0x0 = Used on multicast DSA To_CPU frames with a Code of 0x0 (BPDU/MGMT) and on non-DSA multicast MGMT Control¹ frames.</p> <p>0x1 = Used on DSA To_CPU frames with a Code of 0x1 (Frame to Register Reply). Not used on non-DSA Control frames.</p> <p>0x2 = Used on DSA To_CPU frames with a Code of 0x2 (IGMP/MLD Trap) and on non-DSA Control frames that are IGMP or MLD trapped (Port offset 0x04).</p> <p>0x3 = Used on DSA To_CPU frames with a Code of 0x3 (Policy Trap) and on non-DSA Control frames that are Policy Trapped (Port offset 0x0E).</p> <p>0x4 = Used on DSA To_CPU frames with a Code of 0x4 (ARP Mirror) and on non-DSA Control frames that are ARP Mirrored (Port offset 0x08).</p> <p>0x5 = Used on DSA To_CPU frames with a Code of 0x5 (Policy Mirror) and on non-DSA Control frames that are Policy Mirrored (Port offset 0x0E). (Reserved on the 88E6350R/88E6350 devices)</p> <p>0x6 = Used on DSA To_CPU frames with a Code of 0x6 (Reserved). Not used on non-DSA Control frames.</p> <p>0x7 = Used on unicast DSA To_CPU frames with a Code of 0x0 (unicast MGMT) and on non-DSA unicast MGMT Control² frames.</p> <p>0x8 = Used on DSA From_CPU frames. Not used on non-DSA Control frames.</p> <p>0x9 = Used on DSA Cross Chip Flow Control frames. Not used on non-DSA Control frames.</p> <p>0xA = Used on DSA Cross Chip Egress Monitor frames. Not used on non-DSA Control frames.</p> <p>0xB = Used on DSA Cross Chip Ingress Monitor frames. Not used on non-DSA Control frames.</p> <p>0xC = Used on normal network ports (FrameMode = 0x0, Port offset 0x04) on frames whose EtherType matches the port's PortEType register. Not used on DSA Control frames.</p> <p>0xD = Used on Non-DSA Control frames that contain a Broadcast destination address. Not used on DSA Control frames.</p>

Table 160: Priority Override Table
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
11:8 (cont.)	Pointer	RWT	0xE = Used on Non-DSA Control frames that contain an Ethertype that matches 0x8863 (i.e., PPPoE frames). Not used on DSA Control frames. 0xF = Used on Non-DSA Control frames that contain an Ethertype that matches 0x0800 with a VER = 0x4 or an Ethertype that matches 0x86DD with a VER = 0x6 (i.e., IPv4 or IPv6 frames). Not used on DSA Control frames.
7	QpriAvbEn	RWR	When this entry's bit is set to a one the DataAvb bits below are used to override the frame's Qpri used for AVB enabled ports (AVB Port offset 0x0) or AvbOverride enabled ports (AVB Port offset 0x0). If this bit is cleared to a zero no QpriAvb override will occur for this entry. Up to sixteen QpriAvb override entries are possible in the table. What each entry is used for is defined in the Pointer bits above. Care is needed when setting this bit to a one. When a QpriAvb entry is enabled it will be applied to all frames of the entry's Type. The value in the entry's DataAvb bits will be the Qpri used when mapping all of these frames into an AVB enabled port as this data becomes the final QpriAvb for these frames.
6	Reserved	RES	Reserved for future use.
5:4	DataAvb	RWR	Queue Priority Override Data to use on AVB ports. A value of 0x3 places a frame in the highest priority egress queue. A value of 0x0 places a frame in the lowest priority egress queue. When a frame enters a port its Type is determined (in priority order ³ if it could be multiple Types) and the frame's Type is used to access this table. If the Type's QpriAvbEn bit (bit 7 above) is set to a one then the frame's QpriAvb will be overridden with the value found in this Data field ⁴ . NOTE: These bits are accessible only when the FPriSet bit above equals zero.
3	QPriEn or FPriEn	RWR Except for entry 0x0 & 0x7 in the QPri Set	When FPriSet (bit above) equals zero and when this entry's bit is set to a one the lower two Data bits below are used to override the frame's QPri used for non-AVB ports. If this bit is cleared to a zero no QPri override will occur for this entry. When FPriSet (bit above) equals one and when this entry's bit is set to a one the Data bits below are used to override the frame's FPri ⁵ . If this bit is cleared to a zero no FPri override will occur for this entry. Up to sixteen QPri and FPri override entries are possible in the table. What each entry is used for is defined in the Pointer bits above.



Link Street®

88E6321/88E6320 Functional Specification

Table 160: Priority Override Table
Offset: 0x0F or Decimal 15

Bits	Field	Type	Description
2:0	Data	RWR Except for entries 0x0 & 0x7 which are set to 0x3 in the QPri Set.	Priority Override Data. When FPriSet (bit above) equals zero the lower two bits of this field (bits 1:0) are the entry's Queue Priority Override data (in this case the upper bit, bit 2, can be written as any value and the bit is undefined on reads). A value of 0x3 places a frame in the highest priority egress queue. A value of 0x0 places a frame in the lowest priority egress queue. When FPriSet (bit above) equals one all three bits of this field are the entry's Frame Priority Override data. When a frame enters a port its Type is determined (in priority order ⁶ if it could be multiple Types) and the frame's Type is used to access this table. If the Type's QPrien bit (bit 3 above) is set to a one then the frame's QPri will be overridden with the value found in this Data field ⁷ .

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

Table 161: EEPROM Command
Offset: 0x14 or Decimal 20

Bits	Field	Type	Description
15	EEBusy	SC	EEPROM Unit Busy. This bit must be set to a one to start an EEPROM operation (see EEOp below). Only one EEPROM operation can be executing at one time so this bit must be zero before setting it to a one. When the requested EEPROM operation completes this bit will automatically be cleared to a zero. The transition of this bit from a one to a zero can be used to generate an interrupt (the EEInt in Global 1, offset 0x00).
14:12	EEOp	RWR	EEPROM Opcode. The device supports the following EEPROM operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Reserved 010 = Reserved 011 = Write EEPROM at Addr (Data from EEPROM Data register below) 100 = Read EEPROM from Addr (Data returned in EEPROM Data register below) 101 = Reserved 110 = Restart Register Load execution at Addr ¹ 111 = Reserved
11	Running	RO	Register Loader Running. This bit is set to a one whenever the register loader is busy executing the instructions contained in the EEPROM. An EEPROM Read or Write can only be done when this bit is zero.
10	WriteEn	RO	EEPROM Write Enable. This bit being a one indicates that writing to the EEPROM is possible. If this bit is a zero the Write EEPROM EEOp above will not do anything. This bit reflects the value of the EEE_WP configuration pin after Reset.
9:8	Reserved	RES	Reserved for future use.
7:0	Addr	RWR	EEPROM Address. This is the EEPROM's address where the EEOp (above) is performed and whenever the EEPROM stops executing it contains the address of the last EEPROM command executed..

1. The EEBusy bit will clear as soon as the Register Loader restarts such that this command interface can be available to Hart the EEPROM again. The chip's internal reset is not re-activated by this operation (this is like the chip's current ReLoad except the starting address can be non-zero).

Table 162: EEPROM Data
Offset: 0x15 or Decimal 21

Bits	Field	Type	Description
15:0	Data	RWR	Data to be written to the EEPROM or data that was read back from the EEPROM. The EEPROM action (read or write) and the location of the action (the address inside the EEPROM) are defined in the EEPROM Control Command register above.



Link Street®

88E6321/88E6320 Functional Specification

Table 163: AVB Command Register
Offset: 0x16 or Decimal 22

Bits	Field	Type	Description
15	AVBBusy	SC	<p>AVB unit Busy. This bit must be set to a one to start an AVB operation (see AVBOp below).</p> <p>Only one AVB operation can be executing at one time so this bit must be zero before setting it to a one. When the requested AVB operation completes, this bit will automatically be cleared to a zero.</p>
14:12	AVBOp	RWR	<p>AVB unit Operation code. The devices support the following AVB operations (all of these operations can be executed while frames are transiting through the switch):</p> <ul style="list-style-type: none">000 = No Operation001 = Reserved010 = Reserved011 = Write to the register pointed by AVBAddr below. AVBData registers content gets written into the selected register.100 = Read from the register pointed to by AVBAddr below. The data read from the selected register is transferred to AVBData register.101 = Reserved110 = Read with post increment of register address defined in bits AVBAddr bits below. For PTP data structures, this command instructs the hardware to take a snap-shot of 4 consecutive data registers starting with the AVBAddr location. This is used for capturing time counter values which are more than 16 bits wide along with the sequence identifier information.111 = Reserved
11:8	AVBPort	RWR	<p>This indicates the physical port of this device. These bits indicate the AVB port that is being accessed in the AVBCommand register.</p> <p>For example, if this field is programmed to a 0x1, it indicates that AVB registers belonging to port number 1 are being accessed. Use a value of 0xF to access the AVB Global registers.</p> <p>This indicates the physical port of this device. These bits indicate the AVB port that is being accessed in the AVBCommand register.</p> <p>For example, if this field is programmed to a 0x1, it indicates that AVB registers belonging to port number 1 are being accessed.</p> <p>To access PTP global registers, set the AVBBlock to 0x0 and set AVBPort to 0xF.</p> <p>To access Time Application Interface (TAI) Global registers, set the AVBBlock to 0x0 and set AVBPort to 0xE.</p> <p>To access AVB Policy global registers, set the AVBBlock to 0x1 and set AVBPort to 0xF.</p> <p>To access Qav global registers, set the AVBBlock to 0x2 and set AVBPort to 0xF.</p>

Table 163: AVB Command Register
Offset: 0x16 or Decimal 22

Bits	Field	Type	Description
7:5	AVBBlock	RWR	<p>This field indicates the block of addresses within the device. For example within the AVB register space, this field selects the block like PTP, SRP and Qav etc.</p> <p>The AVBAddr field below selects the specific address within a selected block.</p> <p>0x0: To select PTP register space (documented in section 1.3.8 p 142) 0x1: To select AVB Policy register space (documented in section 1.3.9 p 168) 0x2: To select Qav register space (documented in section 1.3.10 p 176) 0x3 – 0x7: Reserved for future use.</p> <p>NOTE: Accessing registers in the reserved range of the AVBBlock would return all zero's back for the AVBCommand register.</p>
4:0	AVBAddr		These bits indicate the address bits for the register operation being specified in the AVBOp bits specified above.

Table 164: AVB Data Register
Offset: 0x17 or Decimal 23

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



Link Street®

88E6321/88E6320 Functional Specification

Table 165: SMI PHY Command Register¹
Offset: 0x18 or Decimal 24

Bits	Field	Type	Description
15	SMIBusy	SC	SMI PHY Unit Busy. This bit must be set to a one to start an internal SMI operation on the SMI_PHY pins (see SMIOp below). Only one SMI operation can be executing at one time so this bit must be zero before setting it to a one. When the requested SMI operation completes this bit will automatically be cleared to a zero. If the PPU is disabled this bit clears right away.
14:13	Reserved	RES	Reserved for future use.
12	SMIMode	RWR	SMI PHY Mode bit. This bit is used to define the SMI frame type to generate as follows: 0 = Generate IEEE 802.3 Clause 45 SMI frames 1 = Generate IEEE 802.3 Clause 22 SMI frames
11:10	SMIOp	RWR	SMI PHY Opcode. These bits are used to select the SMI opcode to operate on during SMI commands as follows: When the SMIMode bit = 1 then SMIOp = (IEEE 802.3 Clause 22): 00 = Reserved 01 = Write Data Register 10 = Read Data Register 11 = Reserved When the SMIMode bit = 0 then SMIOp = (IEEE 802.3 Clause 45): 00 = Write Address Register 01 = Write Data Register 10 = Read Data Register with post increment on the Address Register 11 = Read Data Register
9:5	DevAddr	RWR	SMI PHY Device Address bits. These bits are used to select the SMI device (Clause 22) or port (Clause 45) to operate on during SMI commands.
4:0	RegAddr	RWR	SMI PHY Register Address bits. These bits are used to select the SMI register (Clause 22) or device class (Clause 45) to operate on during SMI commands.

1. This register can be used to access the PHY registers only when the PPU is enabled (global offset 0x04).

Table 166: SMI PHY Data Register¹
Offset: 0x19 or Decimal 25

Bits	Field	Type	Description
15:0	SMIData	RWR	SMI PHY Data register. During SMI Writes these bits must be written with the SMI PHY data to be written prior to starting the SMI PHY operation (i.e., before setting SMIBusy to a one). During SMI PHY Reads these bits will contain the SMI PHY data that was read after the SMI PHY read operation completes (i.e., SMIBusy returns to a zero). Writes to this register must not be done while SMIBusy is a one. If the PPU is disabled this data will be 0xFFFF.

1. This register can be used to access the PHY registers only when the PPU is enabled (global offset 0x04).

Table 167: Scratch and Misc. Register
Offset: 0x1A or Decimal 26

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 will be loaded into the Scratch and Misc. Control register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	<p>Pointer to the desired octet of Scratch and Misc (Miscellaneous). These bits select one of the possible Scratch and Misc registers for both read and write operations. A write operation occurs if the Update bit is a one (the Scratch and Misc registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found</p> <p>there is placed in the Data bits below (the Scratch and Misc register can be read by first writing to this register, with Update = 0, and then reading this register).</p> <p>The Pointer bits are used to access the Index registers as follows:</p> <ul style="list-style-type: none"> 0x00 to 0x01: Scratch Bytes 0x06 to 0x06: Voltage Regulator control 0x06 0x07 to 0x0A: Reserved for future use 0xB to 0x0F: Energy Efficient Ethernet (EEE) register space 0x10 to 0x1F: Reserved for future use 0x20 to 0x3F: GPIO Port Stall Vectors 0x60 to 0x6F: GPIO registers data and configuration 0x70 to 0x7F: CONFIG reads All other addresses are reserved for future use.
7:0	Data	RWR	Scratch and Misc. Control data read or written to the register pointed to by the Pointer bits. See Table 168 through Table 223 .

Table 168: Scratch Byte 0, Register Index: 0x00 of Scratch and Misc. Control

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

Table 169: Scratch Byte 1, Register Index: 0x01 of Scratch and Misc. Control

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



Link Street®

88E6321/88E6320 Functional Specification

Table 170: Voltage Regulator Control, Register Index: 0x06 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	V1.5sel	RWS to 0x3	1.5 Volt Regulator Selection range. The internal 1.5V regulator's output voltage can be adjusted as follows: 000 = 1.35v 001 = 1.40v 010 = 1.45v 011 = 1.50v (default) 100 = 1.55v (not recommended) 101 = 1.60v (not recommended) 110 = Reserved for future use 111 = Reserved for future use
3:0	V1.1sel	RWS to 0xA	1.1 Volt Regulator Selection range. The internal 1.1V regulator's output voltage can be adjusted as follows: 0000 = 0.825v 0001 = 0.850v 0010 = 0.875v 0011 = 0.900v 0100 = 0.925v 0101 = 0.950v 0110 = 0.975v 0111 = 1.000v 1000 = 1.025v (default) 1001 = 1.050v 1010 = 1.075v 1011 = 1.100v (default) 1100 = Reserved for future use 1101 = Reserved for future use 1110 = Reserved for future use 1111 = Reserved for future use

Table 171: EEE Timer Rates, Register Index: 0x0B of Scratch and Misc. Control

Bits	Field	Type	Description
7:6	Reserved	RES	Reserved for future use.
5:4	AsrtRate	RWR	Assertion Timer Rate. These bits determine the resolution of the Assertion Timer at register index 0x0D as follows: 00 = 1 uSec steps (default) 01 = 1 mSec steps 10 = 1 Sec steps 11 = Reserved
3:2	WakeRate	RWR	Wake Timer Rate. These bits determine the resolution of the Wake Timer at register index 0x0E as follows: 00 = 1 uSec steps (default) 01 = 1 mSec steps 10 = 1 Sec steps 11 = Reserved
1:0	TxIdleRate	RWS to 0x1	TxIdle Timer Rate. These bits determine the resolution of the Tx Idle Timer at register index 0x0F as follows: 00 = 1 uSec steps 01 = 1 mSec steps (default) 10 = 1 Sec steps 11 = Reserved

Table 172: EEE Wake Timer GE, Register Index: 0x0C of Scratch and Misc. Control

Bits	Field	Type	Description
7:0	WakeTime GE	RWS to 0x11	EEE Wake Time for Gigabit Ethernet ports. In order for PHY to exit low power mode and return back to normal mode, a wake up period is needed. These bits determine the time that switch must wait for the PHY to completely exit low power mode before the switch can start to transmit a packet. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by WakeRate in Index 0x0B above. The default value of 0x11 is 17uSec.

Table 173: EEE Assertion Timer, Register Index: 0x0D of Scratch and Misc. Control

Bits	Field	Type	Description
7:0	AsrtTime	RWR	EEE Assertion Time. When the PHY is asked to enter low power mode it cannot immediately leave this mode. This timer determines the time requirement for the PHY to fully power down before a wake up request can be processed. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by AsrtRate in Index 0x0B above. An AsrtTime of 0x0 has the low power idle code deasserted immediately when an idle request is deasserted.



Link Street®

88E6321/88E6320 Functional Specification

Table 174: EEE WakeTimer, Register Index: 0x0E of Scratch and Misc. Control

Bits	Field	Type	Description
7:0	WakeTime	RWS to 0x1E	EEE Wake Time. In order for PHY to exit low power mode and return back to normal mode, a wake up period is needed. These bits determine the time that switch must wait for the PHY to completely exit low power mode before the switch can start to transmit a packet. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by WakeRate in Index 0x0B above. The default value of 0x1E is 30uSec.

Table 175: EEE TxIdle Timer, Register Index: 0x0F of Scratch and Misc. Control

Bits	Field	Type	Description
7:0	TxIdleTime	RWS to 0x02	EEE Transmit Idle Time. These bits determine the time that each TX port must remain idle before telling the PHY to entering its low power state. The value in this register is in micro seconds (uSec), mille seconds (mSec) or seconds as determined by TxIdleRate in Index 0x0B above. If the Transmit Idle Time is set to 0x0 then automatic low power mode is entered as soon as the port's egress queue is empty and has nothing more to send.

Table 176: GPIO 0 Port Stall Vector 0, Register Index: 0x20 of Scratch and Misc. Control

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

Table 177: GPIO 0 Port Stall Vector 1, Register Index: 0x21 of Scratch and Misc. Control

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

Table 178: GPIO 1 Port Stall Vector 0, Register Index: 0x22 of Scratch and Misc. Control

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



Link Street®

88E6321/88E6320 Functional Specification

Table 179: GPIO 1 Port Stall Vector 1, Register Index: 0x23 of Scratch and Misc. Control

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

Table 180: GPIO 2 Port Stall Vector 0, Register Index: 0x24 of Scratch and Misc. Control

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

Table 181: GPIO 2 Port Stall Vector 1, Register Index: 0x25 of Scratch and Misc. Control

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

Table 182: GPIO 3 Port Stall Vector 0, Register Index: 0x26 of Scratch and Misc. Control

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



Link Street®

88E6321/88E6320 Functional Specification

Table 183: GPIO 3 Port Stall Vector 1, Register Index: 0x27 of Scratch and Misc. Control

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

Table 184: GPIO 4 Port Stall Vector 0, Register Index: 0x28 of Scratch and Misc. Control

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

Table 185: GPIO 4 Port Stall Vector 1, Register Index: 0x29 of Scratch and Misc. Control

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

Table 186: GPIO 5 Port Stall Vector 0, Register Index: 0x2A of Scratch and Misc. Control

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



Link Street®

88E6321/88E6320 Functional Specification

Table 187: GPIO 5 Port Stall Vector 1, Register Index: 0x2B of Scratch and Misc. Control

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

Table 188: GPIO 6 Port Stall Vector 0, Register Index: 0x2C of Scratch and Misc. Control

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

Table 189: GPIO 6 Port Stall Vector 1, Register Index: 0x2D of Scratch and Misc. Control

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

Table 190: GPIO 7 Port Stall Vector 0, Register Index: 0x2E of Scratch and Misc. Control

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



Link Street®

88E6321/88E6320 Functional Specification

Table 191: GPIO 7 Port Stall Vector 1, Register Index: 0x2F of Scratch and Misc. Control

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

Table 192: GPIO 8 Port Stall Vector 0, Register Index: 0x30 of Scratch and Misc. Control

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

Table 193: GPIO 8 Port Stall Vector 1, Register Index: 0x31 of Scratch and Misc. Control

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

Table 194: GPIO 9 Port Stall Vector 0, Register Index: 0x32 of Scratch and Misc. Control

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

Table 195: GPIO 9 Port Stall Vector 1, Register Index: 0x33 of Scratch and Misc. Control

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



Link Street®

88E6321/88E6320 Functional Specification

Table 196: GPIO 10 Port Stall Vector 0, Register Index: 0x34 of Scratch and Misc. Control

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

Table 197: GPIO 10 Port Stall Vector 1, Register Index: 0x35 of Scratch and Misc. Control

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

Table 198: GPIO 11 Port Stall Vector 0, Register Index: 0x36 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 11 Port Stall Vector	RWR	GPIO 11's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 11's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 11's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 11 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 11 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is deasserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 199: GPIO 11 Port Stall Vector 1, Register Index: 0x37 of Scratch and Misc. Control

Bits	Field	Type	Description
7	GPIO 11 Port Stall En	RWR	GPIO 11's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 11 pin, if the pin is currently a GPIO input, is used to activate GPIO 11's Port Stall Vector (Index 0x36 above). When this bit is cleared to a zero GPIO 11's Port Stall Vector function is disabled.
6	GPIO 11 Port Stall Value	RWR	GPIO 11's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 11 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 11 Port Stall En bit (above) is set to a one, the GPIO 11 Port Stall Vector (Index 0x36 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 11 Port Stall Vector – Index 0x36 above).
5:0	Reserved	RES	Reserved for future use.

Table 200: GPIO 12 Port Stall Vector 0, Register Index: 0x38 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 12 Port Stall Vector	RWR	GPIO 12's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 12's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 12's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 12 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 12 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is deasserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.



Link Street®

88E6321/88E6320 Functional Specification

Table 201: GPIO 12 Port Stall Vector 0, Register Index: 0x39 of Scratch and Misc. Control

Bits	Field	Type	Description
7	GPIO 12 Port Stall En	RWR	GPIO 12's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 12 pin, if the pin is currently a GPIO input, is used to activate GPIO 12's Port Stall Vector (Index 0x38 above). When this bit is cleared to a zero GPIO 12's Port Stall Vector function is disabled.
6	GPIO 12 Port Stall	RWR	GPIO 12's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 12 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 12 Port Stall En bit (above) is set to a one, the GPIO 12 Port Stall Vector (Index 0x38 above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 12 Port Stall Vector – Index 0x38 above).
5:0	Reserved	RES	Reserved for future use.

Table 202: GPIO 13 Port Stall Vector 0, Register Index: 0x3A of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 13 Port Stall Vector	RWR	GPIO 13's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 13's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 13's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 13 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 13 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is deasserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 203: GPIO 13 Port Stall Vector 1, Register Index: 0x3B of Scratch and Misc. Control

Bits	Field	Type	Description
7	GPIO 13 Port Stall En	RWR	GPIO 13's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 13 pin, if the pin is currently a GPIO input, is used to activate GPIO 13's Port Stall Vector (Index 0x3A above). When this bit is cleared to a zero GPIO 13's Port Stall Vector function is disabled.
6	GPIO 13 Port Stall Value	RWR	GPIO 13's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 13 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 13 Port Stall En bit (above) is set to a one, the GPIO 13 Port Stall Vector (Index 0x3A above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 13 Port Stall Vector – Index 0x3A above).
5:0	Reserved	RES	Reserved for future use.

Table 204: GPIO 14 Port Stall Vector 0, Register Index: 0x3C of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO 14 Port Stall Vector	RWR	GPIO 14's Port Stall Vector. The value in this register is sent to the transmit portion of all the ports in the device when GPIO 14's Port Stall En (below) is set to a one and the value on the device's GPIO pin matches the value in GPIO 14's Port Stall Value (below) and the GPIO pin is a GPIO input. The assertion of this vector to the ports persists as long as the function is enabled (i.e., GPIO 14 Port Stall En equals a one) and as long as the value on the pin matches the programmed stall value (GPIO 14 Port Stall Value) and as long as the pin remains a GPIO input. When a port's bit is set to a one in this vector, that port or ports will stop transmitting their next frame (the current frame is not affected) until this vector is deasserted. Port 0 is controlled by bit 0, port 1 by bit 1, etc.

Table 205: GPIO 14 Port Stall Vector 1, Register Index: 0x3D of Scratch and Misc. Control

Bits	Field	Type	Description
7	GPIO 14 Port Stall En	RWR	GPIO 14's Port Stall Enable. When this bit is set to a one the data that appears of the device's GPIO 14 pin, if the pin is currently a GPIO input, is used to activate GPIO 14's Port Stall Vector (Index 0x3C above). When this bit is cleared to a zero GPIO 14's Port Stall Vector function is disabled.
6	GPIO 14 Port Stall Value	RWR	GPIO 14's Port Stall Value. The value of this bit is used to match the data that appears of the device's GPIO 14 pin, if the pin is currently a GPIO input. If a match occurs and if the GPIO 14 Port Stall En bit (above) is set to a one, the GPIO 14 Port Stall Vector (Index 0x3D above) is presented to the ports to control the transmission of the ports enabled in the vector (see GPIO 14 Port Stall Vector – Index 0x3D above).
5:0	Reserved	RES	Reserved for future use.



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 206: GPIO Configuration, Register Index: 0x60 of Scratch and Misc. Control

Bits	Field	Type	Description
7	GPIO[7] Mode	RWR See Note in Desc.	General Purpose Input Output [7]'s Mode. GPIO[7] is shared with P5_COL. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_COL pin is not needed and if the NO_CPU configuration pin is a zero ¹ , this pin becomes GPIO[7] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[7], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 3 (Register Index 0x6B of Scratch and Misc. Control).
6	GPIO[6] Mode	RO	General Purpose Input Output [6]'s Mode. GPIO[6] is shared with P5_IND[3]. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_IND[3] pin is not needed this pin becomes GPIO[6] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[6], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 3 (Register Index 0x6B of Scratch and Misc. Control).
5	GPIO[5] Mode	RO	General Purpose Input Output [5]'s Mode. GPIO[5] is shared with P5_IND[2]. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_IND[2] pin is not needed this pin becomes GPIO[5] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[5], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 2 (Register Index 0x6A of Scratch and Misc. Control).
4	GPIO[4] Mode	RO	General Purpose Input Output [4]'s Mode. In the 88 pin package GPIO[4] is shared with P5_IND[1]. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_IND[1] pin is not needed, this pin becomes GPIO[4] and this bit will be set to a one, otherwise this bit will be cleared to a zero. In the 108 pin package GPIO[4] is a dedicated pin so this bit will always be a one in this case. When configured to be GPIO[4], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 2 (Register Index 0x6A of Scratch and Misc. Control).

Table 206: GPIO Configuration, Register Index: 0x60 of Scratch and Misc. Control (Continued)

Bits	Field	Type	Description
3	GPIO[3] Mode	RO	<p>General Purpose Input Output [3]'s Mode. In the 88 pin package GPIO[3] is shared with P5_IND[0]. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_IND[0] pin is not needed, this pin becomes GPIO[3] and this bit will be set to a one, otherwise this bit will be cleared to a zero. In the 108 pin package GPIO[3] is a dedicated pin so this bit will always be a one in this case.</p> <p>When configured to be GPIO[3], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 1 (Register Index 0x69 of Scratch and Misc. Control).</p>
2	GPIO[2] Mode	RO	<p>General Purpose Input Output [2]'s Mode. GPIO[2] is shared with P5_INDV. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_INDV pin is not needed this pin becomes GPIO[2] and this bit will be set to a one, otherwise this bit will be cleared to a zero.</p> <p>When configured to be GPIO[2], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 1 (Register Index 0x69 of Scratch and Misc. Control).</p>
1	GPIO[1] Mode	RO	<p>General Purpose Input Output [2]'s Mode. GPIO[2] is shared with P5_INCLK. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_INCLK pin is not needed this pin becomes GPIO[2] and this bit will be set to a one, otherwise this bit will be cleared to a zero.</p> <p>When configured to be GPIO[2], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 1 (Register Index 0x69 of Scratch and Misc. Control).</p>
0	GPIO[0] Mode	RO	<p>General Purpose Input Output [0]'s Mode. GPIO[0] is shared with P5_OUTCLK. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_OUTCLK pin is not needed, this pin becomes GPIO[0] and this bit will be set to a one, otherwise this bit will be cleared to a zero.</p> <p>When configured to be GPIO[0], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x62 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x64 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 0 (Register Index 0x68 of Scratch and Misc. Control).</p>

1. The effect of the NO_CPU configuration pin's value can be inverted. See NormalSMI at index 0x63 of this register.



Link Street®

88E6321/88E6320 Functional Specification

Table 207: GPIO Configuration, Register Index: 0x61 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6	GPIO[14] Mode	RO	General Purpose Input Output [14]'s Mode. GPIO[14] is shared with P6_IND[7]. When Port 6 is configured into a mode (by the P6_MODE pins or by the C_Mode bits in Port offset 0x00) where the P6_IND[7] pin is not needed, this pin becomes GPIO[14] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[14], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 7 (Register Index 0x6F of Scratch and Misc. Control).
5	GPIO[13] Mode	RO	General Purpose Input Output [13]'s Mode. GPIO[13] is shared with P6_IND[6]. When Port 6 is configured into a mode (by the P6_MODE pins or by the C_Mode bits in Port offset 0x00) where the P6_IND[6] pin is not needed, this pin becomes GPIO[13] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[13], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 6 (Register Index 0x6E of Scratch and Misc. Control).
4	GPIO[12] Mode	RO	General Purpose Input Output [12]'s Mode. GPIO[12] is shared with P6_IND[5]. When Port 6 is configured into a mode (by the P6_MODE pins or by the C_Mode bits in Port offset 0x00) where the P6_IND[5] pin is not needed, this pin becomes GPIO[12] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[12], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 6 (Register Index 0x6E of Scratch and Misc. Control).
3	GPIO[11] Mode	RO	General Purpose Input Output [11]'s Mode. GPIO[11] is shared with P6_IND[4]. When Port 6 is configured into a mode (by the P6_MODE pins or by the C_Mode bits in Port offset 0x00) where the P6_IND[4] pin is not needed, this pin becomes GPIO[11] and this bit will be set to a one, otherwise this bit will be cleared to a zero. When configured to be GPIO[11], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control). When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 5 (Register Index 0x6D of Scratch and Misc. Control).

Table 207: GPIO Configuration, Register Index: 0x61 of Scratch and Misc. Control (Continued)

Bits	Field	Type	Description
2	GPIO[10] Mode	RO	<p>General Purpose Input Output [10]'s Mode. GPIO[10] is shared with P6_CRS. When Port 6 is configured into a mode (by the P6_MODE pins or by the C_Mode bits in Port offset 0x00) where the P6_CRS pin is not needed, this pin becomes GPIO[10] and this bit will be set to a one, otherwise this bit will be cleared to a zero.</p> <p>When configured to be GPIO[10], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 5 (Register Index 0x6D of Scratch and Misc. Control).</p>
1	GPIO[9] Mode	RO	<p>General Purpose Input Output [9]'s Mode. GPIO[9] is shared with P6_COL. When Port 6 is configured into a mode (by the P6_MODE pins or by the C_Mode bits in Port offset 0x00) where the P6_COL pin is not needed, this pin becomes GPIO[9] and this bit will be set to a one, otherwise this bit will be cleared to a zero.</p> <p>When configured to be GPIO[9], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 4 (Register Index 0x6C of Scratch and Misc. Control).</p>
0	GPIO[8] Mode	RO	<p>General Purpose Input Output [8]'s Mode. GPIO[8] is shared with P5_CRS. When Port 5 is configured into a mode (by the P5_MODE pins) where the P5_CRS pin is not needed and if the NO_CPU configuration pin is a zero¹, this pin becomes GPIO[8] and this bit will be set to a one, otherwise this bit will be cleared to a zero.</p> <p>When configured to be GPIO[8], the direction of this pin is controlled by the GPIO Direction register (Register Index 0x63 of Scratch and Misc. Control) and the data read and/or write is accessed by the GPIO Data register (Register Index 0x65 of Scratch and Misc., Control).</p> <p>When this bit is a one alternate pin functions can be configured on this pin. See GPIO Pin Control 4 (Register Index 0x6C of Scratch and Misc. Control).</p>

1. The effect of the NO_CPU configuration pin's value can be inverted. See NormalSMI at index 0x63 of this register.



Link Street®

88E6321/88E6320 Functional Specification

Table 208: GPIO Direction, Register Index: 0x62 of Scratch and Misc. Control

Bits	Field	Type	Description
7:0	GPIO Direction [7:0]	RWS to 0x7F	General Purpose Input Output's Direction. This register is used to control the direction of GPIO[7:0]. Bit 0 controls GPIO[0], bit 1 controls GPIO[1], etc. When a GPIO's bit is set to a one that GPIO will become an input. When a GPIO's bit is cleared to a zero that GPIO will become an output. This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x60) above.

Table 209: GPIO Direction, Register Index: 0x63 of Scratch and Misc. Control

Bits	Field	Type	Description
7	NormalSMI	RES but resets to 0x1	Normal SMI vs. GPIO mode. When P5_MODE is not equal to 0x1 or 0x2 the P5_COL and P5_CRS pins are not needed. In this case, when this bit is set to a one, the P5_COL and P5_CRS pins become MDIO_PHY and MDC_PHY, respectively, if the NO_CPU configuration pin was a one during reset. Else the pins become GPIO pins. Clearing this bit to a zero inverts the effect the NO_CPU configuration pin's value has on the function of the P5_COL and P5_CRS pins when they are not needed (i.e., when P5_MODE is not equal to 0x1 or 0x2).
6:0	GPIO Direction [14:8]	RWS to 0x7F	General Purpose Input Output's Direction. This register is used to control the direction of GPIO[14:8]. Bit 0 controls GPIO[8], bit 1 controls GPIO[9], etc. When a GPIO's bit is set to a one that GPIO will become an input. When a GPIO's bit is cleared to a zero that GPIO will become an output. This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x61) above.

Table 210: GPIO Data, Register Index: 0x64 of Scratch and Misc. Control

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

Table 211: GPIO Data, Register Index: 0x65 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:0	GPIO Data [14:8]	RWR	<p>General Purpose Input Output's Data. This register is used to access the data of GPIO[14:8]. Bit 0 accesses GPIO[8], bit 1 accesses GPIO[9], etc. When a GPIO's bit is set to be an input (by the GPIO Direction bits above, Index 0x63) data written to this bit will go to a holding register but will not appear on the pin nor in this register. Reads of this register will return the actual, real-time, data that is appearing on the GPIO's pin.</p> <p>When a GPIO's bit is set to be an output (by the GPIO Direction bits above, Index 0x63) data written to this bit will go to a holding register and will appear on the GPIO's pin. Reads of this register will return the actual, real-time, data that is appearing on the GPIO's pin (which in this case should be the data written, but if its isn't that would be an indication of a conflict).</p> <p>When a pin's direction changes from input to output, the data last written to the holding register appears on the GPIO's pin.</p> <p>This bit only has an affect for GPIO's that are enabled to be GPIO's as noted by their bits being a one in the GPIO Configuration register (Index 0x61).</p> <p>NOTE: Any GPIO that is not enabled (i.e., its Mode bit is zero in Index 0x61) will return a zero on reads.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 212: GPIO Pin Control 0, Register Index: 0x68 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[1] Control	RWR	General Purpose Input Output Interface pin 1 Control. This register is used to control alternate functions of the GPIO[1] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[1] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[1] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[0] Control	RWR	General Purpose Input Output Interface pin 0 Control. This register is used to control alternate functions of the GPIO[0] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[0] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[0] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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

Table 213: GPIO Pin Control 1, Register Index: 0x69 of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[3] Control	RWR	General Purpose Input Output Interface pin 3 Control. This register is used to control alternate functions of the GPIO[3] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[3] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[3] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[2] Control	RWR	General Purpose Input Output Interface pin 2 Control. This register is used to control alternate functions of the GPIO[2] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[2] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[2] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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



Link Street®

88E6321/88E6320 Functional Specification

Table 214: GPIO Pin Control 2, Register Index: 0x6A of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[5] Control	RWR	General Purpose Input Output Interface pin 5 Control. This register is used to control alternate functions of the GPIO[5] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[5] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[5] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[4] Control	RWR	General Purpose Input Output Interface pin 4 Control. This register is used to control alternate functions of the GPIO[4] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[4] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[4] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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

Table 215: GPIO Pin Control 3, Register Index: 0x6B of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[7] Control	RWR	General Purpose Input Output Interface pin 7 Control. This register is used to control alternate functions of the GPIO[7] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[7] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[7] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[6] Control	RWR	General Purpose Input Output Interface pin 6 Control. This register is used to control alternate functions of the GPIO[6] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[6] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[6] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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



Link Street®

88E6321/88E6320 Functional Specification

Table 216: GPIO Pin Control 4, Register Index: 0x6C of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[9] Control	RWR	General Purpose Input Output Interface pin 9 Control. This register is used to control alternate functions of the GPIO[9] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[9] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[9] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[8] Control	RWR	General Purpose Input Output Interface pin 8 Control. This register is used to control alternate functions of the GPIO[8] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[8] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[8] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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

Table 217: GPIO Pin Control 5, Register Index: 0x6D of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[11] Control	RWR	General Purpose Input Output Interface pin 11 Control. This register is used to control alternate functions of the GPIO[11] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[11] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 0 = GPIO[11] 1 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 2 = PTP_EVREQ – Precise Timing Protocol Event Request Input 3 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 4 = SE_RCLK0 – SyncE Receive Clock 0 Output 5 = SE_RCLK1 – SyncE Receive Clock 1 Output 6 = Reserved 7 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[10] Control	RWR	General Purpose Input Output Interface pin 10 Control. This register is used to control alternate functions of the GPIO[10] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[10] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[10] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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



Link Street®

88E6321/88E6320 Functional Specification

Table 218: GPIO Pin Control 6, Register Index: 0x6E of Scratch and Misc. Control

Bits	Field	Type	Description
7	Reserved	RES	Reserved for future use.
6:4	GPIO[13] Control	RWR	General Purpose Input Output Interface pin 13 Control. This register is used to control alternate functions of the GPIO[13] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[13] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[13] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output
3	Reserved	RES	Reserved for future use.
2:0	GPIO[12] Control	RWR	General Purpose Input Output Interface pin 12 Control. This register is used to control alternate functions of the GPIO[12] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[12] Mode bit is a one (Register Index 0x61 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[12] 001 = PTP_TRIGGER – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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

Table 219: GPIO Pin Control 7, Register Index: 0x6F of Scratch and Misc. Control

Bits	Field	Type	Description
7:3	Reserved	RES	Reserved for future use.
2:0	GPIO[14] Control	RWR	General Purpose Input Output Interface pin 14 Control. This register is used to control alternate functions of the GPIO[14] pin when it is not being used as part of some other interface. This register has an effect only if the pin is enabled to be a GPIO pin, i.e., the GPIO[14] Mode bit is a one (Register Index 0x60 of Scratch and Misc., Control). The options are as follows: 000 = GPIO[14] 001 = PTP_TRIG – Precise Timing Protocol Trigger Generate Output 010 = PTP_EVREQ – Precise Timing Protocol Event Request Input 011 = PTP_EXTCLK – Precise Timing Protocol External Clock Input 100 = SE_RCLK0 – SyncE Receive Clock 0 Output 101 = SE_RCLK1 – SyncE Receive Clock 1 Output 110 = Reserved 111 = CLK125 – Free running 125 MHz Clock Output



Note

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



Link Street®

88E6321/88E6320 Functional Specification

Table 220: CONFIG Data0, Register Index: 0x70 of Scratch and Misc. Control

Bits	Field	Type	Description																											
7:0	Config 0	RO	<p>Reset Configuration pin data 0. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values in registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 0's bits are:</p> <table><thead><tr><th>Bit</th><th>CONFIG</th><th>Pin's Primary Name</th></tr></thead><tbody><tr><td>0</td><td>Reserved</td><td></td></tr><tr><td>1</td><td>Reserved</td><td></td></tr><tr><td>2</td><td>Reserved</td><td></td></tr><tr><td>3</td><td>ADDR[0]</td><td>P6_GTXCLK</td></tr><tr><td>4</td><td>ADDR[1]</td><td>P6_OUTD[4]</td></tr><tr><td>5</td><td>ADDR[2]</td><td>P6_OUTD[5]</td></tr><tr><td>6</td><td>ADDR[3]</td><td>P6_OUTD[6]</td></tr><tr><td>7</td><td>ADDR[4]</td><td>P6_OUTD[7]</td></tr></tbody></table> <p>NOTE: The Address CONFIG pins are called ADDR[4:0]n so the value in this register is inverted from what was 'seen' on the pins on the rising edge of RESETn.</p>	Bit	CONFIG	Pin's Primary Name	0	Reserved		1	Reserved		2	Reserved		3	ADDR[0]	P6_GTXCLK	4	ADDR[1]	P6_OUTD[4]	5	ADDR[2]	P6_OUTD[5]	6	ADDR[3]	P6_OUTD[6]	7	ADDR[4]	P6_OUTD[7]
Bit	CONFIG	Pin's Primary Name																												
0	Reserved																													
1	Reserved																													
2	Reserved																													
3	ADDR[0]	P6_GTXCLK																												
4	ADDR[1]	P6_OUTD[4]																												
5	ADDR[2]	P6_OUTD[5]																												
6	ADDR[3]	P6_OUTD[6]																												
7	ADDR[4]	P6_OUTD[7]																												

Table 221: CONFIG Data1, Register Index: 0x71 of Scratch and Misc. Control

Bits	Field	Type	Description																											
7:0	Config 1	RO	<p>Reset Configuration pin data 1. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values in registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 1's bits are:</p> <table><thead><tr><th>Bit</th><th>CONFIG</th><th>Pin's Primary Name</th></tr></thead><tbody><tr><td>0</td><td>LED_SEL[0]</td><td>R1_LED</td></tr><tr><td>1</td><td>LED_SEL[1]</td><td>R2_LED</td></tr><tr><td>2</td><td>NO CPU</td><td>R3_LED</td></tr><tr><td>3</td><td>Reserved</td><td></td></tr><tr><td>4</td><td>Reserved</td><td></td></tr><tr><td>5</td><td>Reserved</td><td></td></tr><tr><td>6</td><td>FLOW</td><td>EE_CLK/C0_LED</td></tr><tr><td>7</td><td>EEE_WP</td><td>P2_OUTEN</td></tr></tbody></table>	Bit	CONFIG	Pin's Primary Name	0	LED_SEL[0]	R1_LED	1	LED_SEL[1]	R2_LED	2	NO CPU	R3_LED	3	Reserved		4	Reserved		5	Reserved		6	FLOW	EE_CLK/C0_LED	7	EEE_WP	P2_OUTEN
Bit	CONFIG	Pin's Primary Name																												
0	LED_SEL[0]	R1_LED																												
1	LED_SEL[1]	R2_LED																												
2	NO CPU	R3_LED																												
3	Reserved																													
4	Reserved																													
5	Reserved																													
6	FLOW	EE_CLK/C0_LED																												
7	EEE_WP	P2_OUTEN																												

Table 222: CONFIG Data2, Register Index: 0x72 of Scratch and Misc. Control

Bits	Field	Type	Description																											
7:0	Config 2	RO	<p>Reset Configuration pin data 2. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values in registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 1's bits are:</p> <table> <thead> <tr> <th>Bit</th> <th>CONFIG</th> <th>Pin's Primary Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>P5_MODE[0]</td> <td>P5_OUTD[0]</td> </tr> <tr> <td>1</td> <td>P5_MODE[1]</td> <td>P5_OUTD[1]</td> </tr> <tr> <td>2</td> <td>P5_MODE[2]</td> <td>P5_OUTD[2]</td> </tr> <tr> <td>3</td> <td>Reserved for future use</td> <td></td> </tr> <tr> <td>4</td> <td>P6_MODE[0]</td> <td>P6_OUTD[0]</td> </tr> <tr> <td>5</td> <td>P6_MODE[1]</td> <td>P6_OUTD[1]</td> </tr> <tr> <td>6</td> <td>P6_MODE[2]</td> <td>P6_OUTD[2]</td> </tr> <tr> <td>7</td> <td>Reserved for future use</td> <td></td> </tr> </tbody> </table>	Bit	CONFIG	Pin's Primary Name	0	P5_MODE[0]	P5_OUTD[0]	1	P5_MODE[1]	P5_OUTD[1]	2	P5_MODE[2]	P5_OUTD[2]	3	Reserved for future use		4	P6_MODE[0]	P6_OUTD[0]	5	P6_MODE[1]	P6_OUTD[1]	6	P6_MODE[2]	P6_OUTD[2]	7	Reserved for future use	
Bit	CONFIG	Pin's Primary Name																												
0	P5_MODE[0]	P5_OUTD[0]																												
1	P5_MODE[1]	P5_OUTD[1]																												
2	P5_MODE[2]	P5_OUTD[2]																												
3	Reserved for future use																													
4	P6_MODE[0]	P6_OUTD[0]																												
5	P6_MODE[1]	P6_OUTD[1]																												
6	P6_MODE[2]	P6_OUTD[2]																												
7	Reserved for future use																													

Table 223: CONFIG Data3, Register Index: 0x73 of Scratch and Misc. Control

Bits	Field	Type	Description																											
7:0	Config 3	RO	<p>Reset Configuration pin data 3. This register returns the values observed after a hardware Reset on the listed CONFIG pins listed below. Some of these CONFIG pins are used to set initial values in registers that are changeable later by software. These register bits will not change in these cases as they always report the value latched for the CONFIG pin after Reset. Config 1's bits are:</p> <table> <thead> <tr> <th>Bit</th> <th>CONFIG</th> <th>Pin's Primary Name</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>P0_S_MODE</td> <td>GPIO[3]</td> </tr> <tr> <td>1</td> <td>P1_S_MODE</td> <td>GPIO[4]</td> </tr> <tr> <td>2</td> <td>LARGE_PKG N/A</td> <td></td> </tr> <tr> <td>3</td> <td>Reserved</td> <td></td> </tr> <tr> <td>4</td> <td>P5_VDDOS[0]</td> <td>P5_OUTEN</td> </tr> <tr> <td>5</td> <td>P5_VDDOS[1]</td> <td>P5_OUTD[3]</td> </tr> <tr> <td>6</td> <td>P6_VDDOS[0]</td> <td>P6_OUTEN</td> </tr> <tr> <td>7</td> <td>P6_VDDOS[1]</td> <td>P6_OUTD[3]</td> </tr> </tbody> </table> <p>NOTE: S_MODE[1:0] are valid in the large package only.</p>	Bit	CONFIG	Pin's Primary Name	0	P0_S_MODE	GPIO[3]	1	P1_S_MODE	GPIO[4]	2	LARGE_PKG N/A		3	Reserved		4	P5_VDDOS[0]	P5_OUTEN	5	P5_VDDOS[1]	P5_OUTD[3]	6	P6_VDDOS[0]	P6_OUTEN	7	P6_VDDOS[1]	P6_OUTD[3]
Bit	CONFIG	Pin's Primary Name																												
0	P0_S_MODE	GPIO[3]																												
1	P1_S_MODE	GPIO[4]																												
2	LARGE_PKG N/A																													
3	Reserved																													
4	P5_VDDOS[0]	P5_OUTEN																												
5	P5_VDDOS[1]	P5_OUTD[3]																												
6	P6_VDDOS[0]	P6_OUTEN																												
7	P6_VDDOS[1]	P6_OUTD[3]																												



Link Street®

88E6321/88E6320 Functional Specification

Table 224: Watch Dog Control Register
Offset: 0x1B or Decimal 27

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 will be loaded into the Watch Dog Control register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14:8	Pointer	RWR	<p>Pointer to the desired octet of Watch Dog. These bits select one of the possible Watch Dog registers for both read and write operations. A write operation occurs if the Update bit is a one (the Watch Dog registers can be written to by writing this register in a single operation). Otherwise a read of the current Pointer occurs and the data found there is placed in the Data bits below (the Watch Dog register can be read by first writing to this register, with Update = 0, and then reading this register).</p> <p>The Pointer bits are used to access the Index registers as follows:</p> <ul style="list-style-type: none">0x00: Watch Dog Interrupt Source0x10 to 0x13: Data Path Watch Dog Interrupts, Masks, Events & History0x40: Auto Fixing Enables <p>All other addresses are reserved for future use.</p>
7:0	Data	RWR	Watch Dog Control data read or written to the register pointed to by the Pointer bits above.

The individual registers accessed by the Watch Dog Control register are described below.

Table 225: Watch Dog Interrupt Source, Register Index: 0x00 of Watch Dog

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

Table 226: Watch Dog Interrupt Source, Register Index: 0x10 of Watch Dog

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD Int (88E6321 only)	RO	Cut Through Watch Dog Interrupt. This bit will be set to a one whenever the CT Watch Dog Mask (Watch Dog index 0x11) is set to a one and the CT Watch Dog Event (Watch Dog index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (Watch Dog index 0x00).
2	QC WD Int	RO	Queue Controller Watch Dog Interrupt. This bit will be set to a one whenever the QC Watch Dog Mask (Watch Dog index 0x11) is set to a one and the QC Watch Dog Event (Watch Dog index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (Watch Dog index 0x00).
1	EgressWD Int	RO	Egress Watch Dog Interrupt. This bit will be set to a one whenever the Egress Watch Dog Mask (Watch Dog index 0x11) is set to a one and the Egress Watch Dog Event (Watch Dog index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (Watch Dog index 0x00).
0	ForceWD Int	RO	Force Watch Dog Interrupt. This bit will be set to a one whenever the Force Watch Dog Mask (Watch Dog index 0x11) is set to a one and the Force Watch Dog Event (Watch Dog index 0x12) is set to a one. This bit being a one will cause the DataPathWDInt bit to be a one (Watch Dog index 0x00).



Link Street®

88E6321/88E6320 Functional Specification

Table 227: Data Path Watch Dog Masks, Register Index: 0x11 of Watch Dog

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD Mask (88E6321 only)	RWS	Cut Through Watch Dog Mask. When this bit is set to a one Cut Through Watch Dog Events (Watch Dog index 0x12) will generate Cut Through Watch Dog interrupts (Watch Dog index 0x10).
2	QC WD Mask	RWS	Queue Controller Watch Dog Mask When this bit is set to a one Queue Controller Watch Dog Events (Watch Dog index 0x12) will generate Queue Controller Watch Dog interrupts (Watch Dog index 0x10).
1	EgressWD Mask	RWS	Egress Watch Dog Mask When this bit is set to a one Egress Watch Dog Events (Watch Dog index 0x12) will generate Egress Watch Dog interrupts (Watch Dog index 0x10).
0	ForceWD Mask	RWS	Force a Watch Dog Mask. When this bit is set to a one Forced Watch Dog Events (Watch Dog index 0x12) will generate Forced Watch Dog interrupts (Watch Dog index 0x10).

Table 228: Data Path Watch Dog Events, Register Index: 0x12 of Watch Dog

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD Event (88E6321 only)	RO	<p>Cut Through Watch Dog Event. If the Cut Through logic detects any CT watch dog issues, this bit will be set to a one, regardless of the setting of the CT Watch Dog Mask bit (Watch Dog index 0x11).</p> <p>This bit is cleared by a SWReset (Global 1 offset 0x04) so it will automatically be cleared to zero if the SWReset on WD bit (Watch Dog index 0x13) is set to a one and this event is unmasked (Watch Dog index 0x11).</p>
2	QC WD Event	RO	<p>Queue Controller Watch Dog Event. If the QC logic detects any QC watch dog issues, this bit will be set to a one, regardless of the setting of the QC Watch Dog Mask bit (Watch Dog index 0x11).</p> <p>This bit is cleared by a SWReset (Global 1 offset 0x04) so it will automatically be cleared to zero if the SWReset on WD bit (Watch Dog index 0x13) is set to a one and this event is unmasked (Watch Dog index 0x11).</p>
1	EgressWD Event	RO	<p>Egress Watch Dog Event. If any port's egress logic detects an egress watch dog issue, this bit will be set to a one, regardless of the setting of the Egress Watch Dog Mask bit (Watch Dog index 0x11).</p> <p>This bit is cleared by a SWReset (Global 1 offset 0x04) so it will automatically be cleared to zero if the SWReset on WD bit (Watch Dog index 0x13) is set to a one and this event is unmasked (Watch Dog index 0x11).</p>
0	ForceWD Event	RWR	<p>Force a Watch Dog Event. When this bit is set to a one a watch dog event is forced as if an enabled watch dog event occurred. This bit allows the testing of software that is designed to service the watch dog events.</p> <p>This bit is cleared by a SWReset (Global 1 offset 0x04) so it will automatically be cleared to zero if the SWReset on WD bit (Watch Dog index 0x13) is set to a one and this event is unmasked (Watch Dog index 0x11).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 229: Data Path Watch Dog History, Register Index: 0x13 of Watch Dog

Bits	Field	Type	Description
7:4	Reserved	RES	Reserved for future use.
3	CT WD History (88E6321 only)	RO	Cut Through Watch Dog History. If the Cut Through logic detects any CT watch dog issues, this bit will be set to a one, regardless of the setting of the CT Watch Dog Mask bit (Watch Dog index 0x11). This bit can only be cleared by a hardware reset.
2	QC WD History	RO	Queue Controller Watch Dog History. If the QC logic detects any QC watch dog issues, this bit will be set to a one, regardless of the setting of the QC Watch Dog Mask bit (Watch Dog index 0x11). This bit can only be cleared by a hardware reset.
1	EgressWD History	RO	Egress Watch Dog History. If any port's egress logic detects an egress watch dog issue, this bit will be set to a one, regardless of the setting of the Egress Watch Dog Mask bit (Watch Dog index 0x11). This bit can only be cleared by a hardware reset.
0	SWReset on WD	RWS	SWReset on Watch Dog Event. When this bit is set to a one, any unmasked watch dog events (Watch Dog index 0x11) will automatically reset the switch core's datapath just as if the SWReset bit (Global 1 offset 0x04) was set to a one. The Watch Dog History bits, above will not be cleared by this automatic SWReset. This allows the user to know if any watch dog event ever occurred even if the switch is configured to automatically recover from a watch dog. When this bit is cleared to a zero unmasked watch dog events will not cause a SWReset. The Watch Dog History bits, above and the Watch Dog Ints (Watch Dog index 0x10) will still be set on any unmasked watch dog events, however.

Table 230: Auto Fixing Enables, Register Index: 0x40 of Watch Dog

Bits	Field	Type	Description
7	RMU TImeOut	RWS	<p>Remote Management Timeout. When this bit is set to a one the Remote Management Unit (RMU) will timeout on Wait on Bit commands. If the bit that is being tested has not gone to the specified value after 1 second has elapsed the Wait on Bit command will be terminated and the Response frame will be sent without any further processing.</p> <p>When this bit is cleared to a zero the Wait on Bit command will wait until the bit that is being tested has changed to the specified value.</p>
6:4	Reserved	RES	Reserved for future use.
3:0	DMPort	RWS to 0xF	Duplex Mismatch Port. When a Duplex Mismatch interrupt event occurs (Global 2 offset 0x00) these bits will reflect the port that generated the interrupt. Software needs to set these bits back to 0xF in order to re-arm the Duplex Mismatch interrupt event so that future Duplex Mismatch events can be reported.



Link Street®

88E6321/88E6320 Functional Specification

Table 231: QoS Weights Register
Offset: 0x1C or Decimal 28

Bits	Field	Type	Description
15	Update	SC	Update Data. When this bit is set to a one the data written to bits 7:0 will be loaded into the QoS Weights octet register selected by the Pointer bits below. After the write has taken place this bit self clears to zero.
14	Reserved	RES	Reserved for future use.
13:8	Pointer	RWR	Pointer to desired octet of QoS Weights. These bits select one of 32 possible QoS Weight Data registers and the QoS Weight Length register for both read and write operations. A write operation occurs if the Update bit is a one. Otherwise a read operation occurs.
7:0	Data	RWS to see text	<p>Octet Data of the programmable QoS Weight table.</p> <p>33 QoS Weight registers are accessed by using the Pointer bits above as follows:</p> <p>0x00 to 0x1F = QoS Weight Table Data. 0x20 = QoS Weight Table Length.</p> <p>The QoS Weight Table Data is a 128 x 2 bit table where each Weight Table Data entry contains four 2-bit entries. Bits 1:0 of the entry at Pointer 0x00 is the 1st table entry. Bits 3:2 are the 2nd entry, etc. The 5th entry is bits 1:0 at Pointer 0x01. The two-bit wide entries are used to contain the desired queue processing priority order (starting with bits 1:0 at Pointer 0x00).</p> <p>The QoS Weight Table Length register is used to define the length of the QoS Weight Table Data. Writing to this register causes the new table to be used by the Queue Controller (so the data at pointers 0x00 to 0x1F must be written 1st).</p> <p>The hardware reset values of this table default to an 8, 4, 2, 1 weight as follows:</p> <p>0x00 = 0x7B (this defines a 3, 2, 3, 1 order) 0x01 = 0x3B (this defines a 3, 2, 3, 0 order) 0x02 = 0x7B (this defines a 3, 2, 3, 1 order) 0x03 = 0x3B (this defines a 3, 2, 3 order) 0x04 to 0x1F = 0x00 0x20 = 0x0F (this indicates the 1st 15 steps in the table are to be used)</p>

Table 232: Misc Register
Offset: 0x1D or Decimal 29

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	5 Bit Port	RWR	<p>Use 5 bits for Port data in the Port VLAN Table (PVT). When this bit is set to a one the 9 bits used to access the PVT memory is:</p> <p>Addr[8:5] = Source Device[3:0] or Device Number[3:0] Addr[4:0] = Source Port/Trunk[4:0]</p> <p>When this bit is cleared to a zero the 9 bits used to access the PVT memory is:</p> <p>Addr[8:4] = Source Device[4:0] or Device Number[4:0] Addr[3:0] = Source Port/Trunk[3:0]</p>
13	NoEgr Policy	RWR	<p>No Egress Policy. When this bit is set to a one Egress 802.1Q Secure and Check discards are not performed. This mode allows a non-802.1Q enabled port to send a frame to an 802.1Q enabled port that is configured in the Secure or Check 802.1Q mode (see Port offset 0x08). In this situation the frames will egress even if the VID assigned to the frame is not found in the VTU.</p> <p>When this bit is cleared to zero and the Egress port's 802.1Q mode is Secure or Check (see Port offset 0x08) the VID assigned to all frames mapped to this port must be found in the VTU or the frame will not be allowed to egress this port.</p>
12:0	Reserved	RES	Reserved for future use.



Link Street®

88E6321/88E6320 Functional Specification

Table 233: Misc Register
Offset: 0x1F or Decimal 31

Bits	Field	Type	Description
15:13	Reserved	RES	Reserved for future use.
12:8	Cut Through Hold (88E6321 only)	RWR	<p>Cut Through Burst Hold amount. To support bursts of frames in Cut Through mode, once a Cut Through connection is made between ports, the Cut Through connection needs to be held beyond the end of each transmitted frame. This hold time keeps the last Cut Through connection active until the next ingressing frame can be processed to see if it is also to be Cut Through.</p> <p>If the Cut Through Hold time is too small the Cut Through fabric will disconnect before the next ingressing frame can be Cut Through allowing any Store and Forward frames that are queued up to egress the port.</p> <p>If the Cut Through Hold time is too long the port will be idle longer than it needs to be before switching to the Store and Forward queues and transmitting any frames that may be stored there.</p> <p>The Cut Through Hold register determines the number of octets a Cut Through connection is held after the last bytes of a frame's CRC is transmitted. The default value of 0x00 breaks the connection right away. A value of 0x16 (22 decimal) will hold the connection for a 96-bit IFG + 64-bit Preamble with 2 bytes of pad in case the IFG expanded due to PPM clock differences.</p>
7:0	Reserved	RES	Reserved for future use.

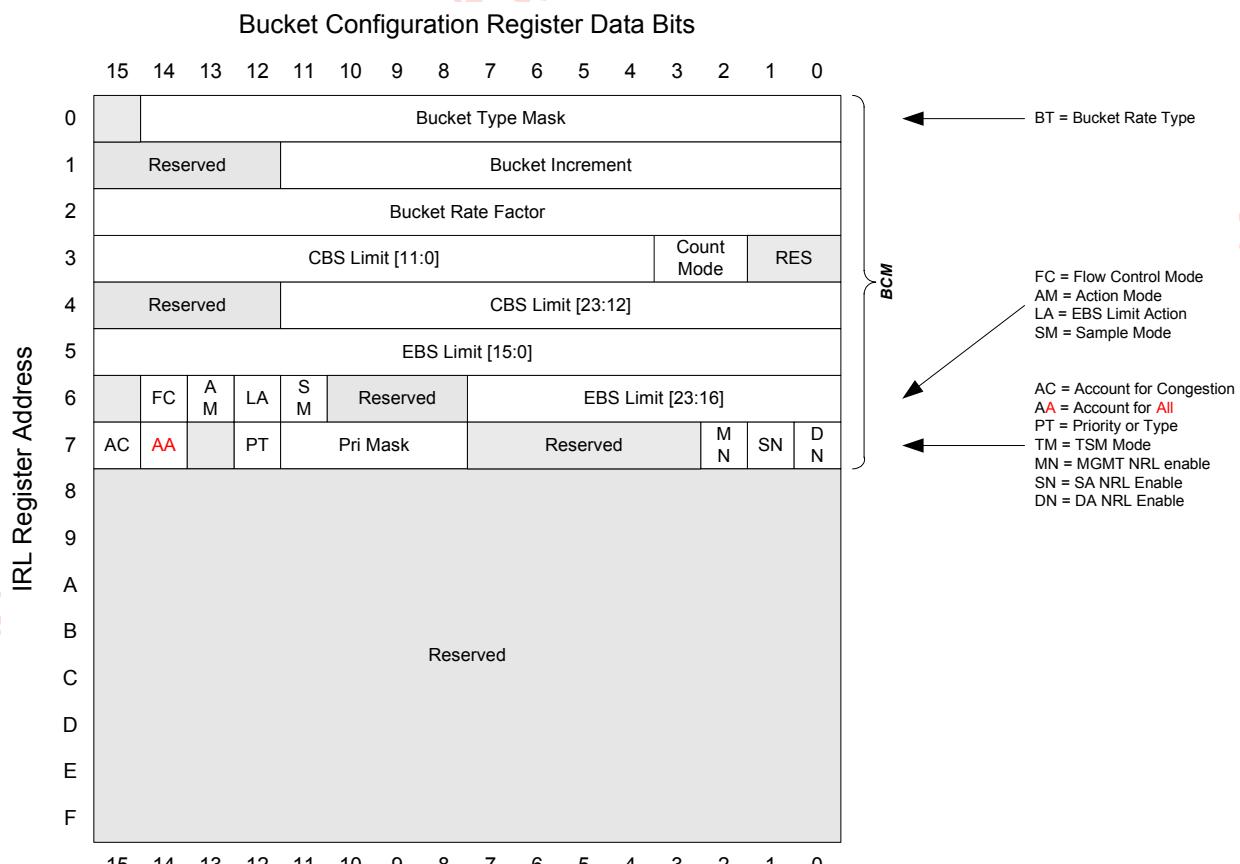
10.5

Port Ingress Rate Limiting (PIRL) Registers

The device contains a set of Port Ingress Rate Limiting (PIRL) registers that effect the selected Ethernet ports in the device. Each PIRL register is 16-bits wide and their bit assignments are shown in Table 6, below.

There are five sets of these registers per port, one set per PIRL resource or bucket. These registers are accessible by Global 2 registers at offsets 0x09 and 0x0A (Ingress Rate Command register and Ingress Rate Data register).

Figure 66: PIRL Register bit Map (from Global 2 offsets 0x09 and 0x0A)



A detailed description of the PIRL registers follow.



Link Street®

88E6321/88E6320 Functional Specification

Table 234: PIRL Bucket Configuration Register
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use. NOTE:
14:0	BktType Mask	RWR	This field has the following definition if BktRateType = 0; [0] – Unknown Unicast The definition of unknown unicast in the context of PIRL is that if the MAC DA search resulted in a failure and if the ingress parsing engine does not classify the frame as either PolMirror or PolTrap. If the ingress parsing engine did mark a frame as an unknown and either PolMirror or PolTrap, then such packets would be tracked based on bits 13 and 14 of BktTypeMask field. [1] – Unknown Multicast The definition of unknown multicast in the context of PIRL is that if the MAC DA search resulted in a failure and if the ingress parsing engine does not classify the frame as either PolMirror or PolTrap. If the ingress parsing engine did mark a frame as an unknown and either PolMirror or PolTrap, then such packets would be tracked based on bits 13 and 14 of BktTypeMask field. [2] – Broadcast [3] – Multicast [4] – Unicast [5] – MGMT Frames [6] – Reserved [7] – ARP [8] – Flow 0 or TCP Data [9] – Flow 1 or TCP Ctrl (if any of the TCP FLAGS[5:0] bits are set) [10] – Flow 2 or UDP [11] – Flow 3 or NON_TCPUDP (covers IGMP, ICMP, GRE, IGRP, L2TP) [12] – IMS (Ingress Monitor Source) [13] – PolicyMirror [14] – PolicyTrap Flow [3:0] are selected for bits 11 to 8 of this register when TCAM Flows = 1 (PIRL Bucket Configuration, offset 0x07). One of the Flow [3:0] wires will be set to a one when the TCAM FlowID is valid for this frame ¹ and FlowID[5] is zero. FlowID[1:0] determine which Flow [3:0] wire is set to a one, where Flow 0 will be a 1 if FlowID[1:0] = 0, Flow 1 will be a 1 if FlowID[1:0] = 1, etc. If FlowID[5] = 1 then none of the Flow [3:0] wires will be set to a one such that no PIRL matches on them can be made. NOTE: Flow[3:0] are only valid on the 88E6320 device.

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

Table 235: PIRL Bucket Configuration Register
Offset: 0x1 or Decimal 1

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
11:0	Bkt Increment	RWR	<p>Bucket increment.</p> <p>This parameter indicates the amount of tokens that need to be added for each byte of frame information.</p>

Table 236: PIRL Bucket Configuration Register
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
15:0	BktRate Factor	RWR	<p>Bucket Rate Factor.</p> <p>This is a factor which determines the amount of tokens that need to be decremented for each rate resource decrement (which is done periodically based on the Committed Information Rate).</p> <p>The higher the value of this field the higher will be the decrement value.</p> <p>If SMode is programmed to a one, then BktRateFactor is expected to be programmed to a zero.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 237: PIRL Bucket Configuration Register
Offset: 0x3 or Decimal 3

Bits	Field	Type	Description
15:4	CBSLimit [11:0]	RWR	Committed Burst Size limit (lower 12 bits). This indicates the committed information burst amount. The upper bits of this register is at IRL offset 0x4 (the next register below).
3:2	Count Mode	RWR	<p>Frame bytes to be accounted for in the rate resource's rate limiting calculations.</p> <p>The supported configurations of this field (bits 1 down to 0) are:</p> <ul style="list-style-type: none">00 = Frame based01 = Count all Layer1 bytes10 = Count all Layer2 bytes11 = Count all Layer3 bytes <p>Frame based configures the rate limiting resource to account for the number of frames from a given port mapped to this rate resource.</p> <p>Layer1 = Preamble (8 Bytes) + Frame's DA to CRC + IFG (12 bytes) Layer2 = Frame's DA to CRC Layer3 = Frame's DA to CRC – 18¹ – 4 (if frame is tagged²)</p> <p>A frame is considered tagged if it is either Customer or Provider tagged during ingress.</p>
1:0	Reserved	RES	Reserved for future use.

1. The 18 bytes are: 6 for DA, 6 for SA, 2 for EtherType and 4 for CRC.
2. Only one tag is counted even if the frame contains more than one tag (i.e., it is Provider tagged).

Table 238: PIRL Bucket Configuration Register
Offset: 0x4 or Decimal 4

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11:0	CBSLimit [23:12]	RWR	Committed Burst Size limit (upper 12 bits). This indicates the committed information burst amount. The lower bits of this register is at IRL offset 0x3 (the next register above).

Table 239: PIRL Bucket Configuration Register
Offset: 0x5 or Decimal 5

Bits	Field	Type	Description
15:0	EBSLimit	RWR	<p>Excess Burst Size limit (lower 16 bits). The upper bits of this register is at IRL offset 0x6 (the next register below).</p> <p>If the ingress rate resources' BktTokenCount exceeds the EBSLimit, based on the equation {EBSLimit – BktTokenCount < CBSLimit} EBSLimitAction is taken on the incoming frame.</p> <p>NOTE: If ActionMode=1, the BktTokenCount can exceed EBSLimit and if the ActionMode = 0, the BktTokenCount is clamped at EBSLimit.</p>

Table 240: PIRL Bucket Configuration Register
Offset: 0x6 or Decimal 6

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14	PirlFC Mode	RWR	<p>Port Ingress Rate Limit Flow Control Mode. This bit is used to determine when the flow control (asserted due to ingress rate limiting threshold exceeding reasons) gets deasserted.</p> <p>0 = De-assert flow control when ingress rate resource has become empty i.e., the ingress rate resource can accept more frames as it is empty.</p> <p>1 = De-assert flow control when ingress rate resource has enough room to accept at least one frame of size determined by the value programmed in the CBS_Limit field as specified in PIRL offset 0x3 & 0x4.</p> <p>For example, if the CBS_Limit for the ingress rate resource is programmed to be 2k Bytes, then the flow control will get de-asserted if there is at least 2k Bytes worth of tokens available in the ingress rate resource.</p>
13	Action Mode ¹	RWR	<p>PIRL Action mode.</p> <p>When enabled (by setting this bit to one) configures this rate limiting resource to accept an incoming frame even though there are not enough tokens to accept the entire incoming frame.</p> <p>When disabled (by clearing this bit to zero) configures this rate limiting resource to take an action specified in the EBSLimitAction if there is not enough tokens available to accept the entire incoming frame.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 240: PIRL Bucket Configuration Register (Continued)

Offset: 0x6 or Decimal 6

Bits	Field	Type	Description
12	EBSLimit Action	RWR	<p>If the incoming port information rate exceeds the EBS_Limit, this field specifies the action that needs to be taken for the violating traffic.</p> <p>0 = indicates that the frame that was received on the port that this particular rate resource is assigned to will get discarded if the EBSLimit has been exceeded.</p> <p>1 = indicates that a flow control frame could get sent back to the source if the flow control is enabled for that port and if EBSlimit has been exceeded.</p> <p>NOTE: If any of the PIRL resource buckets for this port were to discard the packet the packet would get discarded by the switch and similarly if any of the PIRL resource buckets for this port were to send a flow control packet back to the source the flow control packets does get sent.</p>
11	SMode	RWR	<p>Sampling Mode.</p> <p>This mode is used for sampling one out of so many frames / bytes (determined by the configured rate resource parameters) for a stream of frames (determined by the IMS or PolMirror bin in the BktTypeMask configuration) that are being monitored. The stream could be identified by the ingress engine as a Policy mirror and packet sampling can be applied for that stream using one of the rate resources.</p> <p>In this mode, once the rate resource's EBSLimit is exceeded, the next incoming frame from this port that is assigned to this resource gets sent out to the mirror destination. After sending a sample frame, the token count within the rate resource is reset to zero and the bucket increments continue for each subsequent frame arrival.</p> <p>When this bit is set to a one, the sampling mode is enabled and by clearing this bit to a zero the sampling mode is disabled.</p> <p>The sampling mode is useful for limiting the number of Mirror frames sent to the mirror destination or for sampling any of the frame types defined in the BktTypeMask field (RateType = 0) or for sampling frames from a given port (if RateType is programmed to 1).</p> <p>In the device, since there are five rate resources per port and given that each of these rate resources can be programmed to track different types of traffic streams with different bucket limits, if any of the bucket's logic were to decide that the frame needs to be discarded then the frame would not get Sampled to the mirror destination.</p> <p>When TSMMode is 0x1, this field is reserved.</p>
10:8	Reserved	RES	Reserved for future use.

Table 240: PIRL Bucket Configuration Register (Continued)
Offset: 0x6 or Decimal 6

Bits	Field	Type	Description
7:0	EBSLimit	RWR	<p>Excess Burst Size limit (upper 8 bits). The lower bits of this register is at IRL offset 0x5 (the next register above).</p> <p>If the ingress rate resources' BktTokenCount exceeds the EBSLimit, based on the equation {EBSLimit – BktTokenCount < CBSLimit} EBSLimitAction is taken on the incoming frame.</p> <p>NOTE: If ActionMode=1, the BktTokenCount can exceed EBSLimit and if the ActionMode = 0, the BktTokenCount is clamped at EBSLimit. When TSMMode is 0x1, the EBSLimit bits 23:16 are reserved.</p>

1. This bit is expected to be enabled for TCP based applications and disabled for media streaming kind of applications where timing of the data is critical.

Table 241: PIRL Bucket Configuration Register
Offset: 0x7 or Decimal 7

Bits	Field	Type	Description
15	AcctForQCong	RWR	<p>Account for queue congestion discards.</p> <p>When enabled (by setting this bit to one), this bit indicates that even if the incoming frames are discarded in the chip because of output port queue congestion, the ingress rate limiting logic accounts for the incoming bytes (or frames if CountMode is configured to be 0x0) in the bucket_increment calculations.</p> <p>When disabled (by clearing this bit to zero), this bit indicates that if the incoming frames are discarded in the chip because of output port queue congestion, the ingress rate limiting logic does not account for the incoming bytes (or frames if CountMode is configured to be 0x0) in the bucket_increment calculations.</p>
14	AcctForAll	RWR	<p>Account for all frames.</p> <p>When enabled (by setting this bit to one), this bit indicates to the rate limiting logic that even if the incoming frames are filtered in the chip because of ingress policy reasons, or they are discarded due to errors or Port States, account for the incoming bytes (or frames if the CountMode is configured to be 0x0) anyway in the bucket_increment calculations. Basically, this mode counts all the bytes of data that came 'down the wire' into the switch port.</p> <p>If SMode = 1, AcctForAll enabling would allow the ingress policy filtered frames which are classified by the ingress block as PolMirror to the mirror destination.</p> <p>When disabled (by clearing this bit to zero), this bit indicates to the rate limiting logic that if the incoming frames are filtered in the chip because of ingress policy reasons, or they are discarded due to errors or Port States, do not account for the incoming bytes (or frames if the CountMode is configured to be 0x0) in the bucket_increment calculations. Basically, this mode counts the frames that were accepted into the switch memory.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 241: PIRL Bucket Configuration Register (Continued)

Offset: 0x7 or Decimal 7

Bits	Field	Type	Description
13	Reserved	RES	Reserved for future use.
12	PriOrPT	RWR	<p>Priority Or Packet Type.</p> <p>When this bit is set to a one, the frame types defined in the BktTypeMask field (see below) OR the priority bits defined in PriMask (see below) field, determine the incoming frames that get rate limited using this ingress rate resource.</p> <p>When this bit is cleared to a zero, the frame types defined in the BktTypeMask field AND the priority bits defined in PriMask field determine the incoming frames that get rate limited using this rate resource.</p> <p>For example if BktTypeMask[4] = 1 (i.e., unicast frames) and PriMask[3:0] = 0x4 (priority 2 frames), if PriOrPT is set to a one, then either unicasts or Priority 2 frames are accounted for in the ingress limiting calculations for this rate resource. If PriOrPT is cleared to a zero, then all unicast frames that are classified to be priority2 frames are accounted for in the ingress limiting calculations for this rate resource.</p>
11:8	PriMask	RWR	<p>Priority Mask.</p> <p>Each bit indicates one of the four queue priorities. Setting each one of these bits indicates that this particular rate resource is slated to account for incoming frames with the enabled bits' priority.</p> <p>For example, if bits zero and two are set i.e., this field is set to 0x5, frames with queue priority of zero and two are accounted for in this ingress rate resource.</p> <p>NOTE: If PriOrPT bit affects if all frames with a certain priority get rate limited using this rate resource or not (refer to the PriOrPT field description).</p> <p>If this field is set to 0x0, priority of the frame doesn't have any affect on the ingress rate limiting calculations done for this ingress rate resource.</p>
7:4	Reserved	RES	Reserved for future use.
3	TCAM Flows Valid for 88E6321	RWR	<p>Enable the TCAM for Flow identifiers for Flows 0 to 3. When this bit is set to a one, bits 11:8 of the BktTypeMask are connected to the TCAM's FlowID Action bits. In this way, up to four flows per port can be limited and/or counted.</p> <p>When this bit is a zero BktTypeMask bits 11:8 mean NON_TCPUDP, UDP, TCP Ctrl and TCP Data, respectively.</p>

Table 241: PIRL Bucket Configuration Register (Continued)
Offset: 0x7 or Decimal 7

Bits	Field	Type	Description
2	MGMT NrlEn	RWR	<p>Management Non Rate Limit Enable.</p> <p>When this bit is cleared to a zero all frames that are classified by the ingress frame classifier as MGMT frames would be considered to be ingress rate limited as far as this particular ingress rate resource is concerned.</p> <p>When this bit is set to a one, all frames that are classified as MGMT frames by the ingress frame classifier would be excluded from the ingress rate limiting calculations for this particular ingress rate resource.</p> <p>In trusted¹ network environments, MGMT frames could be passing through the switches purely for network administration and management of the switches. In such scenarios it may be necessary to not consider MGMT frames as part of the end customers traffic. This bit provides an option to either consider or not consider MGMT frames as part of the ingress rate limiting for a given rate resource.</p>
1	SANrlEn	RWR	<p>SA Non Rate Limit enable. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one then SA ATU non rate limiting overrides can occur on this port. An SA ATU non rate limiting override occurs when the source address of a frame results in an ATU hit where the SA's MAC address returns an EntryState that indicates Non Rate Limited². When this occurs the frame will not be ingress rate limited.</p>
0	DANrlEN	RWR	<p>DA Non Rate Limit enable. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one then DA ATU non rate limiting overrides can occur on this port. A DA ATU non rate limiting override occurs when the destination address of a frame results in an ATU hit where the DA's MAC address returns an EntryState that indicates Non Rate Limited. When this occurs the frame will not be ingress rate limited.</p>

1. Trusted networks in this context are those which have customers connected to the switch that are never expected to generate DoS attacks using MGMT frames.
2. SA Non Rate Limiting Override can only occur on MAC addresses that are Static or where the Port is Locked, and where the port is the mapped source port for the MAC address.



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

10.6

AVB Registers

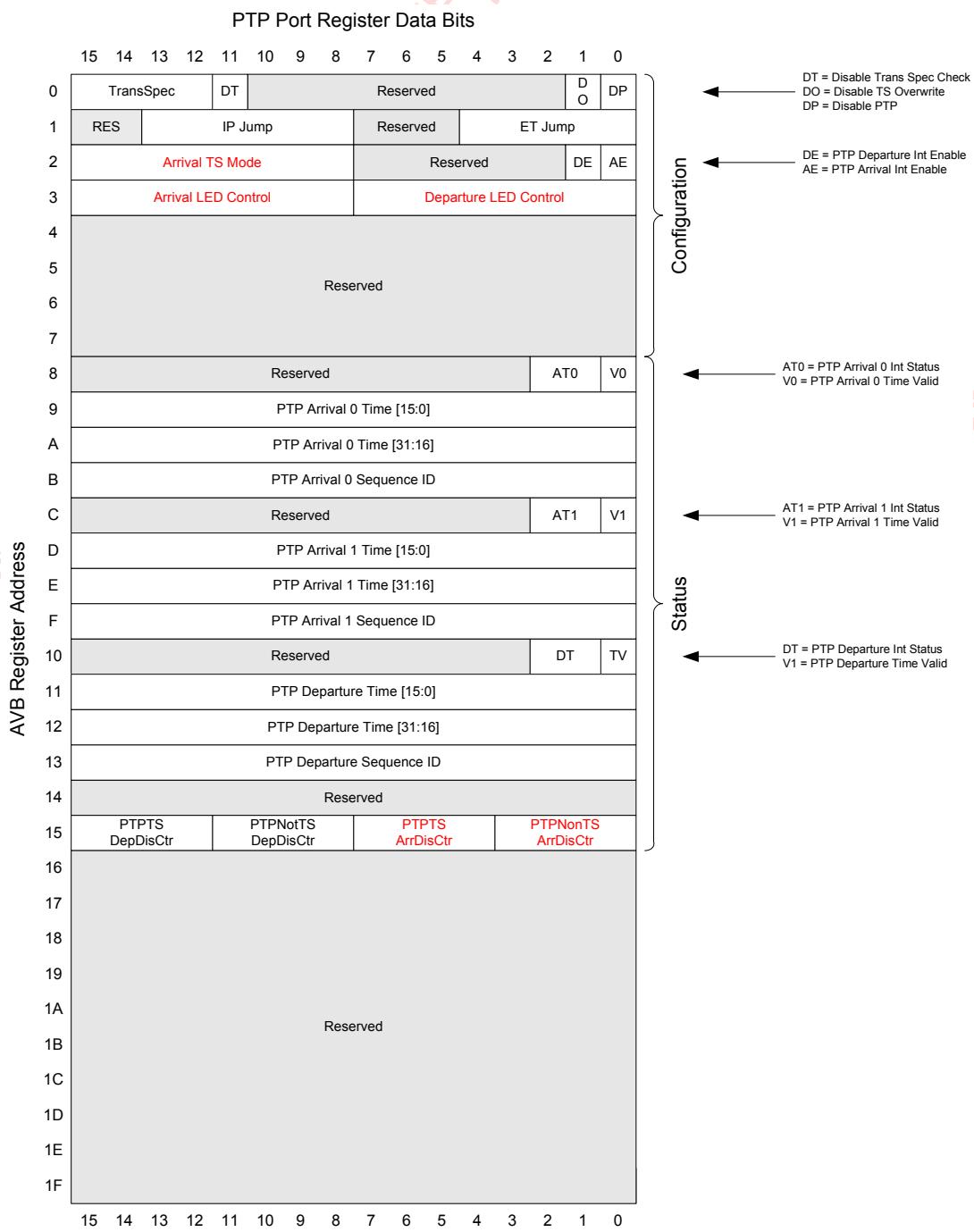
The 88E6321/88E6320 device contains global registers that affect all the Audio Video Bridging (AVB) functions of the device. These registers are accessed using AVBCommand and AVBData registers. The Global 2 AVB registers are used to access the following AVB blocks by using various AVBBlock values (Global 2 offset 0x16):

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

10.6.1 Precise Timing Protocol (PTP) Registers

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

Figure 67: PTP Global Configuration Data Structure Registers





Link Street®

88E6321/88E6320 Functional Specification

Table 242: PTP Port Config Register
Offset: 0x0 or Decimal 0

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

Table 243: PTP Port Config Register
Offset: 0x1 or Decimal 1

Bits	Field	Type	Description
15:14	Reserved	RES	Reserved for future use.
13:8	IPJump	RWS	<p>Internet Protocol Jump added to ETJump below, points to the start of the frame's IP Version byte (assuming it is not 802.1Q tagged). Frames with 802.1Q tags are automatically detected.</p> <p>This field specifies to the PTP hardware logic how many bytes should it skip starting at the value specified by ETJump (bits 4:0). Note that this specifies the jump to the beginning of the IPv4 or IPv6 headers for the hardware parser.</p> <p>This allows flexibility in the hardware to skip past the protocol chains that are specific to customer networks including MPLS etc.</p> <p>For example if ETJump is programmed to 0xC and IPJump is programmed to 0x16, this indicates the hardware to skip 0x22 bytes in order to get to the IP header. It can either be IPv4 or IPv6 header.</p> <p>NOTE: A value of 0x0 or 0x1 in this register is invalid and is not supported.</p>
7:5	Reserved	RES	Reserved for future use.
4:0	ETJump	RWS	<p>EtherType Jump points to the start of the frame's EtherType (assuming it is not 802.1Q tagged). Frames with 802.1Q tags are automatically detected.</p> <p>This field specifies to the PTP hardware logic how many bytes should it skip starting from MAC-DA of the frame to get to the EtherType of the frame. The hardware would skip so many bytes and then compare the next 2 bytes to the configured PTPEType (PTP Global Configuration Register 0, Page 8, Register 0).</p> <p>This allows flexibility in the hardware to skip past the protocol chains that are specific to customer networks including DSA-Tag, IEEE802.1q tag, Provider tag etc.</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 244: PTP Port Config Register
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
15:8	ArrTSMode	RWR	<p>Arrival Time Stamp Mode. This field is used to configure the Arrival Time Stamp frame modification mode as follows: 0x00: Arrival Time Stamp frame modification is disabled. 0x01: Add the PTPArr0Time associated with enabled PTP Event frames at the end of the frame increasing the frame's size by four bytes. 0x02 to 0x03: Reserved for future use. 0x04 to 0xEF: Overwrite the PTPArr0Time associated with enabled PTP Event frames into the frame without increasing the frame's size. The location in the frame where the PTPArr0Time is placed is controlled by this register. It is placed ArrTSMode bytes past the start of the PTP Common Header. For example, to place the time stamp in the four Reserved bytes in the PTP Common Header, set this register to a value of 0x10. If the end of the frame is reached prior to the completion of this overwrite, the PTPArr0Time is placed at the end of the frame increasing the frame's size by enough bytes for it to fit. 0xF0 to 0xFF: Reserved for future use.</p>
7	FilterAct	RWR	<p>Filter all or most of the 802.1 Protocol frames from the Port's Activity LEDs. When this bit is set to a one all 802.1 protocol frames (those with a DA = 01:C2:80:00:00:0x) will be potentially filtered from the port's Activity LED as determined by the ArrLEDCtrl and DepLEDCtrl registers (PTP Port offset 0x03). When this bit is cleared to a zero only the 802.1 gPTP protocol frames will be potentially filtered from the port's Activity LED.</p> <p>Note: This bit affects the port's LEDs only. The frames themselves are not filtered inside the switch because of this bit.</p>
6:2	Reserved	RES	Reserved for future use.
1	PTPDepInt En	RWR	<p>Precise Time Protocol Port Departure Interrupt enable.</p> <p>This field indicates the per-port interrupt enable for outgoing PTP frame from a given port. When a bit is enabled in this field it indicates that whenever hardware logic time stamps a PTP frame to this port, it needs to send an interrupt to the CPU.</p> <p>0x0: Disable PTP departure counter based interrupt generation. Even if the PTPDepTimeValid is set to 0x1, PTPInt doesn't get generated by hardware logic for outgoing PTP frames.</p> <p>0x1: Enable PTP departure counter based interrupt generation. If the PTPDepTimeValid is set to 0x1, PTPInt does get generated by hardware logic for outgoing PTP frames.</p> <p>Hardware logic only time stamps the PTP frames when configured to do so by MsldITSEn field (specified above).</p>

Table 244: PTP Port Config Register (Continued)
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
0	PTPArrInt En	RWR	<p>Precise Time Protocol Port Arrival Interrupt enable.</p> <p>This field indicates the per-port interrupt enable for incoming PTP frames from a given port. When a bit is enabled in this field it indicates that whenever hardware logic time stamps a PTP frame from this port, it needs to send an interrupt to the CPU.</p> <p>0x0: Disable PTP arrival counter based interrupt generation. Even if the PTPArr0TimeValid or PTPArr1TimeValid is set to 0x1 for that port, PTPInt doesn't get generated by hardware logic for incoming PTP frames from this port.</p> <p>0x1: Enable PTP arrival counter based interrupt generation. If the PTPArr0TimeValid or PTPArr1TimeValid is set to 0x1, PTPInt does get generated by hardware logic for incoming PTP frames.</p> <p>Hardware logic only time stamps the PTP frames when configured to do so by MsIDTSEn field (specified above).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 245: PTP Port Config Register
Offset: 0x3 or Decimal 3

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

Table 246: PTP Port Status Register
Offset: 0x8 or Decimal 8

Bits	Field	Type	Description
15:3	Reserved	RES	Reserved for future use.
2:1	PTPArr0Int Status	RWR	<p>Precise Time Protocol Arrival Time 0 Interrupt Status</p> <p>The PTP Arrival time 0 Interrupt bit gets set for a given port when an incoming PTP frame is time stamped in PTPArr0Time counter.</p> <p>0x0 = Normal i.e., none of the error conditions stated below are valid for this packet.</p> <p>0x1 = If the PTPArr0Time counter with its associated valid and SequenceID got overwritten as there were more than one PTP frame that needed to use arrival0 counters arrived into the switch through this port before CPU cleared the corresponding valid and counter bits for the previous PTP frame(s).</p> <p>0x2 = If the incoming frame couldn't be time stamped in hardware because the DisTSOverwrite was set to a 0x1 and PTPArr0TimeValid was 0x1 when this PTPFrame was processed in hardware logic. This can happen when there is more than one PTP frame that needs time stamping into arrival 0 counters arrives into the switch core before CPU clears the valid bits for the previous frame.</p> <p>0x3 = Reserved</p> <p>NOTE: If the PTP frame gets discarded inside the switch for policy, CRC, queue congestion or any other reasons then one of the PTP arrival discard counters get updated (PTPNonTSArrDisCtr or PTPTSArrDisCtr). See the discard counter description for further details.</p>
0	PTPArr0 TimeValid	RWR	<p>Precise Time Protocol Arrival 0 Time Valid</p> <p>When the PTPArr0Time value is updated by hardware, this bit is set to a 0x1 validating the time counter.</p> <p>0x0: PTPArr0Time is not valid.</p> <p>0x1: PTPArr0Time is valid and PTPArr0IntStatus represents the status information for the PTPArr0Time counter. Note that this is set by hardware for the frames which are assured to reach the CPU. For frames with CRC error etc., this bit will not be set but either PTPNonTSArrCtr or PTPTSArrCtr is updated.</p> <p>NOTE: This valid bit needs to be cleared by software after reading the value and hardware doesn't provide any auto-clearing mechanisms. This is because hardware has no way to figure out if software is done reading all the relevant registers for a particular Time counter before clearing the valid bit.</p>



Link Street®

88E6321/88E6320 Functional Specification

**Note**

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

Table 247: PTP Port Status Register

Offset: 0x9 or Decimal 9

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

Table 248: PTP Port Status Register
Offset: 0xA or Decimal 10

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

Table 249: PTP Port Status Register
Offset: 0xB or Decimal 11

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



Link Street®

88E6321/88E6320 Functional Specification

Table 250: PTP Port Status Register
Offset: 0xC or Decimal 12

Bits	Field	Type	Description
15:3	Reserved	RES	Reserved for future use.
2:1	PTPArr1Int Status	RWR	<p>Precise Time Protocol Arrival Time 1 Interrupt Status</p> <p>The PTP Arrival time 1 Interrupt bit gets set for a given port when an incoming PTP frame is time stamped in PTPArr1Time counter.</p> <p>0x0 = Normal i.e., none of the error conditions stated below are valid for this packet.</p> <p>0x1 = If the PTPArr1Time counter with its associated valid and SequenceID got overwritten as there were more than one PTP frame that needed to use arrival1 counters arrived into the switch through this port before CPU cleared the corresponding valid and counter bits for the previous PTP frame(s).</p> <p>0x2 = If the incoming frame couldn't be time stamped in hardware because the DisTSOverwrite was set to a 0x1 and PTPArr1TimeValid was 0x1 when this PTPFrame was processed in hardware logic. This can happen when there is more than one PTP frame that needs time stamping into arrival 1 counters arrives into the switch core before CPU clears the valid bits for the previous frame.</p> <p>0x3 = Reserved</p> <p>NOTE: If the PTP frame gets discarded inside the switch for policy, CRC, queue congestion or any other reasons then one of the PTP arrival discard counters get updated (PTPNonTSArrDiscCtr or PTPTSArrDiscCtr). See the discard counter description for further details.</p>
0	PTPArr1 TimeValid	RWR	<p>Precise Time Protocol Arrival 1 Time Valid</p> <p>When the PTPArr1Time value is updated by hardware, this bit is set to a 0x1 validating the time counter.</p> <p>0x0: PTPArr1Time is not valid.</p> <p>0x1: PTPArr1Time is valid and PTPArr1IntStatus represents the status information for the PTPArr1Time counter.</p> <p>NOTE: This is set by hardware for the frames which are assured to reach the CPU. For frames with CRC error etc., this bit will not be set but either PTPNonTSArrCtr or PTPTSArrCtr is updated.</p> <p>NOTE: This valid bit needs to be cleared by software after reading the value and hardware doesn't provide any auto-clearing mechanisms. This is because hardware has no way to figure out if software is done reading all the relevant registers for a particular Time counter before clearing the valid bit.</p>



Note

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



Table 251: PTP Port Status Register
Offset: 0xD or Decimal 13

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

Table 252: PTP Port Status Register
Offset: 0xE or Decimal 14

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

Table 253: PTP Port Status Register
Offset: 0xF or Decimal 15

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

Table 254: PTP Port Status Register
Offset: 0x10 or Decimal 16

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



Link Street®

88E6321/88E6320 Functional Specification

Table 254: PTP Port Status Register (Continued)

Offset: 0x10 or Decimal 16

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

**Note**

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

Table 255: PTP Port Status Register
Offset: 0x11 or Decimal 17

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

Table 256: PTP Port Status Register
Offset: 0x12 or Decimal 18

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



Link Street®

88E6321/88E6320 Functional Specification

Table 257: PTP Port Status Register
Offset: 0x13 or Decimal 19

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

Table 258: PTP Port Status Register
Offset: 0x15 or Decimal 21

Bits	Field	Type	Description
15:12	PTPTS DepDisCtr	RWR	Precise Time Protocol Departure frame discard counter for PTP frames that need hardware time stamping. This counter is incremented by the hardware logic whenever it discards a PTP frame that needs hardware time stamping (i.e., PTPEtype is a match and MslddTSEn bit for the corresponding Msgld field in the outgoing PTP frame is set). The PTP frame could be discarded because of CRC reasons in egress pipe. This counter wraps around in hardware.
11:8	PTPNonTS DepDisCtr	RWR	Precise Time Protocol Departure frame discard counter for PTP frames that do not need hardware time stamping. This counter is incremented by the hardware logic whenever it discards a PTP frame that doesn't need to be time stamped (i.e., PTPEtype is a match but MslddTSEn bit for the corresponding Msgld field in the outgoing PTP frame is not set). The PTP frame could be discarded because of CRC reasons in egress pipe. This counter wraps around in hardware.
7:4	PTPTS ArrDisCtr	RWR	Precise Time Protocol arrival frame discard counter for PTP frames that need hardware time stamping. This counter is incremented by the hardware logic whenever it discards a PTP frame that needs hardware time stamping (i.e., PTPEtype is a match and MslddTSEn bit for the corresponding Msgld field in the outgoing PTP frame is set). The PTP frame could be discarded because of CRC, Policy, Queue congestion or any other reason inside the switch core (Policy and Queue congestion discards are detected only if the port is not using the TCAM, i.e., the Port's TCAM Mode = 0x0 in Port offset 0x0D). This counter wraps around in hardware.

Table 258: PTP Port Status Register (Continued)
Offset: 0x15 or Decimal 21

Bits	Field	Type	Description
3:0	PTPNonTS ArrDisCtr	RWR	<p>Precise Time Protocol Non time stamp Arrival frame discard counter.</p> <p>This counter is incremented by the hardware logic when ever it discards a PTP frame that doesn't need hardware time stamping (i.e., PTPEtype is a match but MsIdTSEn bit for the corresponding MsgId field in the outgoing PTP frame is not set). The PTP frame could be discarded because of CRC, Policy, Queue congestion or any other reason inside the switch core (Policy and Queue congestion discards are detected only if the port is not using the TCAM, i.e., the Port's TCAM Mode = 0x0 in Port offset 0x0D).</p> <p>This counter wraps around in hardware.</p>

Figure 68: PTP Global Register bit Map (Global 2 offset 0x16 and 0x17 w/AVBBLock = 0x0 & AVBPort = 0xF)

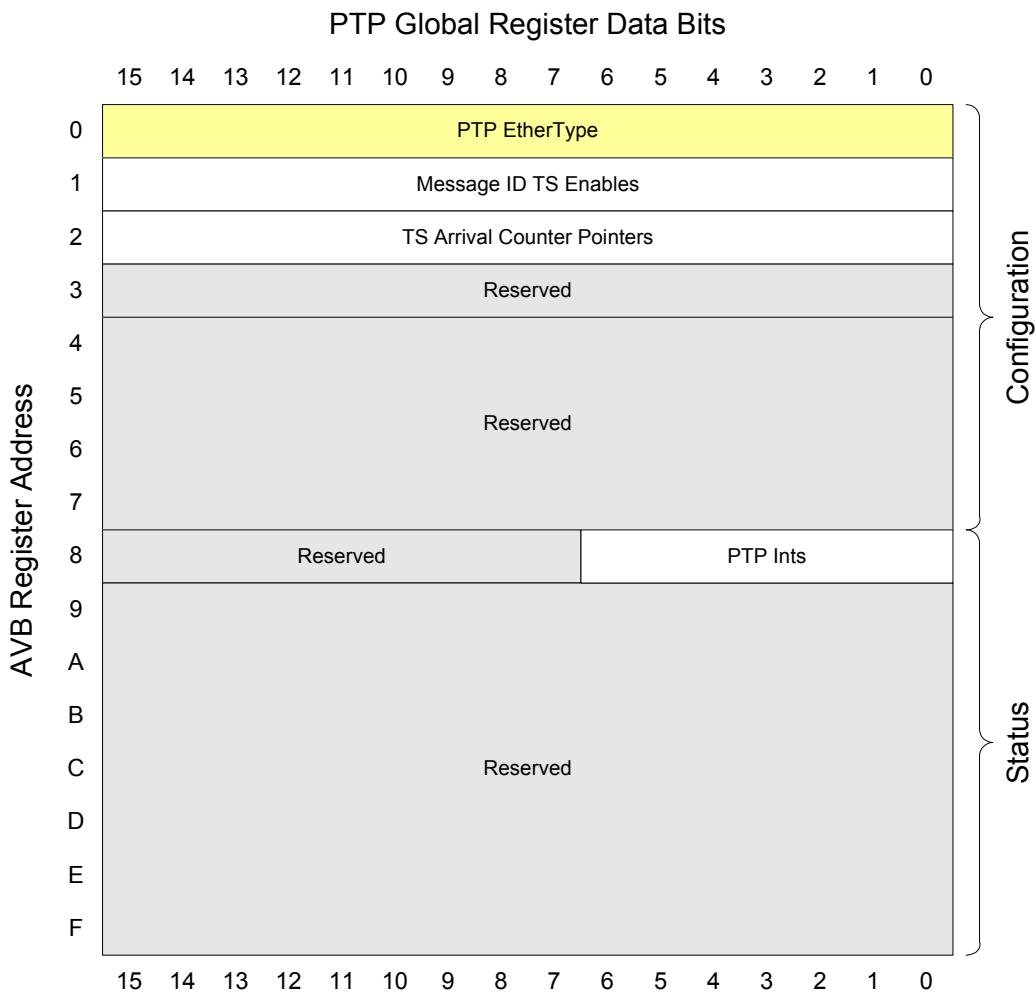


Table 259: PTP Global Config Register, AVBPort = 0xF
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15:0	PTPEType	RWS to 0x88F7	<p>Precision Time Protocol Ether Type.</p> <p>All PTP frames are recognized using a combination of a specific EtherType and MessageID values (part of the PTP Common Header). The actual numeric value is not yet defined in the IEEE802.1AS standard. It is possible that all IEEE1588 time sync frames and IEEE802.11 wireless LAN location estimation time sync messages follow the same EtherType but varying Ether subtypes (aka messageID).</p> <p>The MsgIDTSEn (specified below) qualifies the types of frames that the hardware needs to time stamp.</p>

Table 260: PTP Global Config Register, AVBPort = 0xF
Offset: 0x1 or Decimal 1

Bits	Field	Type	Description
15:0	MsdID TSEn	RWR	<p>Message Identifier Time Stamp Enable.</p> <p>MessageID is part of the PTP common header for time sync frames. There are PTP frames which need to be time stamped by hardware and some that don't need to be. This field identifies the PTP frame types that need to be time stamped by the hardware. Some of the PTP frames may need to be time stamped only when they enter the switch core and not when they are either being sent to or received from the CPU (assuming an external CPU in the system).</p> <p>The MsdIDTSEn refers to the bit mask enables where each bit indicates whether the vectorized¹ MessageID value needs to be time stamped or not.</p> <p>0x0: indicates to hardware to NOT time stamp both incoming and/or outgoing PTP frames which match the MessageID.</p> <p>0x1: indicates to hardware to time stamp both incoming and/or outgoing PTP frames which match the MessageID.</p> <p>For example if MessageID field (in the PTP common header) with a value of 0x4 ought to be time stamped in hardware then MsdIDTSEn[4] should be configured to a 0x1. Then for the incoming PTP frame with the MessageID field of 0x4 one of the two arrival counters get updated (PTPArr0Time or PTPArr1Time. The exact time counter is identified by TSArrPtr field below). For an outgoing PTP frame with the MessageID field of 0x4 TSDepTime counter gets updated for an incoming frame with the MessageID field from the PTPCommon header matching 0x4.</p>

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



Link Street®

88E6321/88E6320 Functional Specification

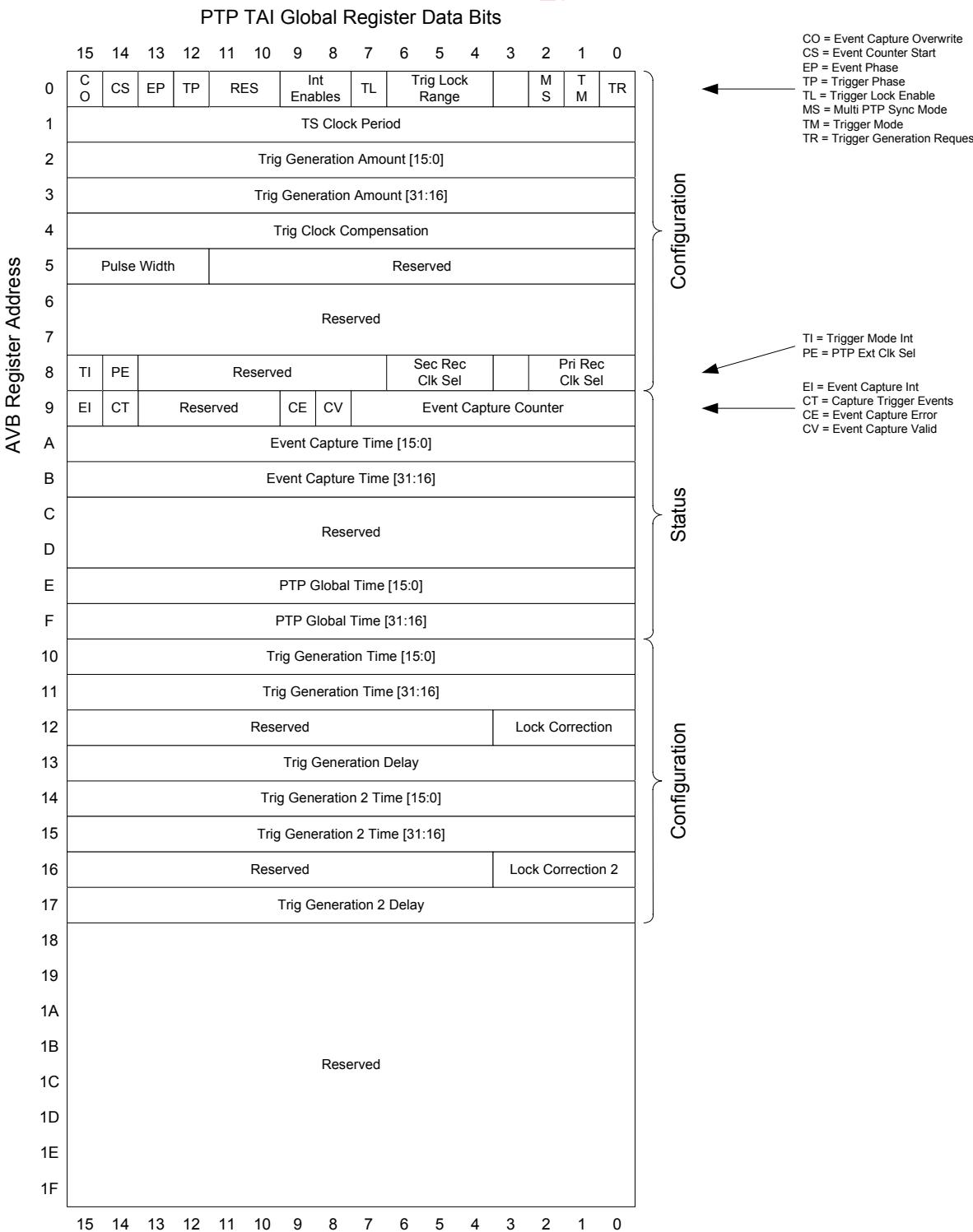
Table 261: PTP Global Config Register, AVBPort = 0xF
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
15:0	TSArrPtr	RWR	<p>Time Stamp Arrival Time Counter Pointer.</p> <p>If the incoming PTP frame needs to be time stamped (based on MsdIDTSEn), this field determines whether the hardware logic should use PTPArr0Time or PTPArr1Time for storing the arriving frames' time stamp information.</p> <p>Each bit in this field corresponds to the sixteen combinations of the vectorized MessageID field. For example if TSArrPtr[2] is set to a 0x1 it indicates to the hardware that if MsdIDTSEn[2] is set then use PTPArr1Time counter for storing the incoming PTP frames' time stamp.</p> <p>On the contrary if TSArrPtr[2] is set to a 0x0 that indicates to the hardware that if MsdIDTSEn[2] is set then use PTPArr0Time counter for storing the incoming PTP frames' time stamp.</p>

Table 262: PTP Global Status Register, AVB = 0xF
Offset: 0x8 or Decimal 8

Bits	Field	Type	Description
15:67	Reserved	RES	Reserved for future use.
5:06:0	PTPInt	RO	<p>Precise Time Protocol Interrupt</p> <p>The PTP Interrupt bit gets set for a given port when an incoming PTP frame is time stamped and PTPArrIntEn for that port is set to 0x1. Similarly PTP Interrupt bit gets set for a given port when an outgoing PTP frame is time stamped and PTPDeplIntEn for that port is set to 0x1.</p> <p>The hardware logic sets this per port bit based on above criteria and gets cleared upon software reading and clearing the corresponding time counter valid bits that are valid for that port.</p> <p>These interrupts are connected to the AVBInt interrupt (Global 1 offset 0x00)..</p>

Figure 69: PTP TAI Global Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x0 & AVBPort = 0xE)





Link Street®

88E6321/88E6320 Functional Specification

Table 263: TAI Global Config Register, AVBPort = 0xE
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15	Event CapOv	RWR	<p>Event Capture Overwrite.</p> <p>When this bit is set to a one it enables overwriting the EventCap registers (TAI offsets 0x09 to 0x0B) whenever an EventReq occurs (see EventPhase below). In this mode the hardware will overwrite the EventCapRegister even if the previously captured event register data has not been read by the software. This mode returns the data from the last EventReq.</p> <p>When this bit is cleared to a zero it configures the hardware to capture a single event, i.e., take a snapshot of PTP Global Timer value at the first EventReq (see EventPhase below) and wait for software to read the EventCapRegister before capturing another event. This mode returns the data from the first EventReq.</p>
14	EventCtr Start	RWR	<p>Event Counter Start.</p> <p>When this bit is set to a one it enables incrementing the EventCapCtr register (TAI offset 0x09) whenever an EventReq occurs (see EventPhase below).</p> <p>When this bit is cleared to zero the EventCapCtr is not modified even when EventReq occurs (see EventPhase below).</p>
13	Event Phase	RWR	<p>Event Phase.</p> <p>When this bit is set to a one an EventReq occurs on the falling edge of the PTP_EVREQ input selected by the GPIO matrix (Global 2 offset 0x1A) or on the trailing edge of PTP_TRIG when internally sampled (see the CaptureTrig bit in TAI offset 0x09).</p> <p>When this bit is cleared to a zero an EventReq occurs on the rising edge of the PTP_EVREQ input selected by the GPIO matrix (Global 2 offset 0x1A) or on the leading edge of PTP_TRIG when internally sampled.</p>
12	Trig Phase	RWR	<p>Trigger Phase.</p> <p>When this bit is set to a one the active phase of the PTP_TRIG output to the GPIO matrix (Global 2 offset 0x1A) is inverted to be active low. For example, the pulse mode of PTP_TRIG will be normally high with a low pulse and the 50% duty cycle's leading edge is the falling edge.</p> <p>When this bit is cleared to a zero the active phase of the PTP_TRIG output to the GPIO matrix (Global 2 offset 0x1A) is normal active high. For example, the pulse mode of PTP_TRIG will be normally low with a high pulse and the 50% duty cycle's leading edge is the rising edge.</p> <p>NOTE: This bit has no effect on the internal phase of PTP_TRIG.</p>
11:10	Reserved	RES	Reserved for future use.

Table 263: TAI Global Config Register, AVBPort = 0xE (Continued)
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
9	TrigGen IntEn	RWR	<p>Trigger Generator Interrupt Enable.</p> <p>When this bit is set to a one the TAI block will generate an interrupt whenever a TrigGen pulse event has occurred.</p> <p>When this bit is cleared to zero no interrupts are generated by the TrigGen logic.</p>
8	EventCap IntEn	RWR	<p>Event Capture Interrupt Enable.</p> <p>When this bit is set to a one the TAI block will generate an interrupt whenever an EventReq occurs.</p> <p>When this bit is cleared to zero no interrupts are generated by the EventCap logic.</p>
7	TrigLock	SC	<p>Trigger Lock. When this bit is set to a one the leading edge of PTP_TRIG (see TrigPhase above) will be adjusted to the value contained in the TrigGenTime register (TAI offsets 0x10 & 0x11) if and only if the leading edge of PTP_TRIG occurs +/- the number of PTP Clocks as defined in the TrigLockRange register below. A correction will also be made to the rising edge of PTP_TRIG2 by setting it to the value contained in the TrigGen2Time register (TAI offsets 0x14 & 0x15) if and only if the leading edge of PTP_TRIG2 occurs +/- the number of PTP Clocks as defined in the TrigLockRange register below.</p> <p>Note: The TrigLockRange, the TrigGenTime and the TrigGen2Time registers must be configured before this bit is set to a one.</p> <p>Once the TrigGenTime and TrigGen2Time are both past in time, this bit will self clear (i.e., it will be active for only one possible correction per wrap around of the 32-bit Global Timer). This bit will clear if the Global time has passed even if a correction was not needed or done.</p> <p>When this bit clears the Lock correction amount, if any, will be registered in the Lock Correction fields for PTP_TRIG and PTP_TRIG2 (TAI offsets 0x12 & 0x16).</p>
6:4	TrigLock Range	RWR	Trigger Locking Range. These bits are used along with the TrigLock bit above. They determine the +/- error limit to adjust and re-center the leading edge of PTP_TRIG in PTP_CLK increments (8ns if using the internal clock) or PTP_EXTCLK increments (if using an external PTP clock).
3	Reserved	RES	Reserved for future use.



Link Street®

88E6321/88E6320 Functional Specification

Table 263: TAI Global Config Register, AVBPort = 0xE (Continued)

Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
2	MultiPTP SyncMode	RWR	<p>Multiple PTP devices sync mode. Used in systems where multiple PTP enabled devices' need to synchronize their PTP Global Time counters (TAI offsets 0x0E & 0x0F).</p> <p>When this bit is set to a one an EventReq (see EventPhase above) transfers the value in TrigGenAmt[31:0] (TAI offsets 0x02 & 0x03) into the PTP Global Time[31:0] register (TAI offsets 0x0E & 0x0F). The EventCapTime[31:0] (TAI offsets 0x0A & 0x0B) is also updated at the same time with the previous value that the PTP Global Time[31:0] register contained prior to be updated.</p> <p>When 0x0, the EventRequest interface operates normally (i.e. an EventReq transfers the value of the PTP Global Time[31:0] register to the EventCapTime[31:0] register based on the setting of the EventCapOv register above).</p>
1	TrigMode	RWR	<p>Trigger Mode. When this bit is set to a one, Pulse mode is enabled. In this mode matches between the PTP Global Timer (TAI offsets 0x0E & 0x0F) and the TrigGenAmt register (TAI offsets 0x02 & 0x03) generate a pulse on the PTP_TRIG signal. The width of the pulse is specified by PulseWidth (TAI offset 0x05). Note: The minimum pulse width that can be generated is one TSCLKPer amount (TAI offset 0x01) and the maximum pulse width is 15 times the TSCLKPer. The phase of the pulse is controlled by TrigPhase bit above.</p> <p>When this bit is cleared to zero the 50% duty cycle clock mode is enabled. In this mode the value specified in the TrigGenAmt is used as the period for generating a 50% duty cycle clock on the PTP_TRIG signal. Note that the minimum clock period that can be generated on the PTP_TRIG signal is 2 times the TSCLKPer amount. The frequency of this clock can be adjusted in ps increments (see TrigClkComp, TAI offset 0x04) and it can be realigned to a specific time (see TrigLock bit above). The first leading edge of the 50% duty cycle clock will occur the first time the PTP Global Time (PTP TAI offsets 0x0E & 0x0F) equals the value in the non-zero TrigGenTime register (PTP TAI offsets 0x10 & 0x11) after TrigGenReq, below, is set to a one. This leading edge control occurs as long as the TrigGenTime register is non-zero. If it is zero the leading edge will occur when the TrigGenReq bit below is set to a one without regard to the PTP Global Time. The phase of the leading edge is controlled by the TrigPhase bit above.</p> <p>Note: When the 50% duty cycle clock is selected, a TrigGen2 clock is also generated with the same frequency and compensations, but with a different leading edge time which is selected with the TrigGen2Time register (PTP TAI offsets 0x14 & 0x15).</p> <p>For example if a 1 pps signal needs to be generated, the TrigMode cleared to zero, and if TSCLKPer is 8ns the TrigGenAmt is set to 125 x 106 cycles.</p>

Table 263: TAI Global Config Register, AVBPort = 0xE (Continued)
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
0	TrigGen Req	RWR or SC	<p>Trigger Generation Request.</p> <p>Trigger Generation Request. When this bit is set to a one, it enables a one-shot pulse or the 50% duty cycle clock generation on PTP_TRIG as previously configured by the TrigGenAmt, TrigMode, TrigClkComp, PulseWidth, TrigGenTime and TrigGenDelay fields.</p> <p>If TrigMode (above) is set to a one (pulse mode) this bit will self clear after the pulse occurs. If TrigMode is cleared to a zero the 50% duty cycle clock will continue running as long as this bit is set to a one.</p>

Table 264: TAI Global Config Register, AVBPort = 0xE
Offset: 0x1 or Decimal 1

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



Link Street®

88E6321/88E6320 Functional Specification

Table 265: TAI Global Config Register, AVBPort = 0xE
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
15:0	TrigGen Amt[15:0]	RWR	<p>Trigger Generation Amount bits 15 to 0 of a 32-bit register. This field specifies the PTP Time Application Interface trigger generation time amount.</p> <p>When TrigMode is set to one, the value specified in this field is compared with the PTP Global Timer (TAI offset 0x09 & 0x0A) and when it matches the first time, a pulse is generated on PTP_TRIG whose width is controlled by PulseWidth (TAI offset 0x05). In this mode there is an internal delay of three TSClkPer before the leading edge of the pulse will be seen on the PTP_TRIG output pin.</p> <p>When TrigMode is cleared to zero, the value is used as a clock period in TSClkPer increments (TAI offset 0x01) to generate an output clock on the PTP_TRIG and PTP_TRIG2 signals (see TrigPhase in TAI offset 0x00). In this mode the TrigClkComp amount (TAI offset 0x04) gets accumulated once per TrigGenAmt cycle and when this accumulated value exceeds the value specified in TSClkPer, one TSClkPer amount gets added to the next trailing edge of PTP_TRIG and of PTP_TRIG2 clock outputs, respectively.</p> <p>The upper 16-bits of this register are contained in the register below.</p>

Table 266: TAI Global Config Register, AVBPort = 0xE
Offset: 0x3 or Decimal 3

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

Table 267: TAI Global Config Register, AVBPort = 0xE
Offset: 0x4 or Decimal 4

Bits	Field	Type	Description
15:0	TrigClk Comp	RWR	<p>Trigger mode Clock Compensation Amount in pico seconds as a signed number. This field is used in 50% duty cycle clock mode only (when TrigMode is cleared to zero and TrigGenReq is set to one).</p> <p>This field specifies the remainder amount for the clock that is being generated with a period specified by the TrigGenAmt (TAI offset 0x02 & 0x03). If the clock this device is using for the PTP hardware clock is slightly too fast then this compensation is a positive number of ps. If the clock is slightly too slow then this compensation is a negative number of ps (accomplished by writing the 2's complement value of ps into this register and then reducing TrigGenAmt (TAI offsets 0x02 & 0x03) by 1).</p> <p>In the 50% duty cycle clock mode this register gets accumulated once per TrigGenAmt cycle and when this accumulated value exceeds the value specified in TSClkPer (TAI offset 0x01), one TSClkPer amount gets added to the next PTP_TRIG and PTP_TRIG2 clock outputs.</p>

Table 268: TAI Global Config Register, AVBPort = 0xE
Offset: 0x5 or Decimal 5

Bits	Field	Type	Description
15:12	Pulse Width	RWS 0xF	<p>Pulse width in units of TSClkPer (TAI offset 0x01). This specifies the width of the pulse that gets generated on PTP_TRIG (see TrigPhase in TAI offset 0x00) when the one shot pulse mode is selected (TrigMode is set to one and TrigGenReq is set to one).</p> <p>NOTE: Setting this register to 0x0 will cause unpredictable results.</p>
11:0	Reserved	RES	Reserved for future use.



Link Street®

88E6321/88E6320 Functional Specification

Table 269: TAI Global Configuration Register, AVBPort = 0xE
Offset: 0x8 or Decimal 8

Bits	Field	Type	Description
15	TrigGen Int	ROC	Trigger generate mode Interrupt. The TrigGenInt bit gets set by the TAI block when the TrigGenIntEn is set to a one (TAI offset 0x00) and when the one shot pulse is generated (TrigMode is set to one in TAI offset 0x00). This interrupt gets tied to the AVBInt interrupt (Global 1 offset 0x00).
14	PtpExtClk	RWR	PTP External Clock select. When this bit is cleared to a zero the PTP core gets its clock from an internal 125 MHz clock based on the device's XTAL_IN input. When this bits is set to a one the PTP core gets its clock from the device's PTP_EXTCLK pin. NOTE: Do not select the PTP_EXTCLK pin unless the pin has a clock and one of the GPIO pins is configured to be the PTP_EXTCLK (Global 2 offset 0x1A indexes 0x68 to 0x6F).
13:7	Reserved	RES	Reserved for future use.
6:4	SecRecClkSel (88E6321 only)	RWS to 0x7	Synchronous Ethernet Secondary Recovered Clock Select. This field indicates the internal PHY number whose recovered clock will be presented on the GPIO pin which is configured to select SE_RCLK1. The reset value of 0x7 selects no clock for SE_RCLK1. The SE_RCLK1 frequency is 25 MHz when SecRecClkSel points to a physical PHY or SERDES (3 for PHY 3, etc.). Selecting a port where there is no PHY (or SERDES) will connect the SE_RCLK1 signal to the internal 25 MHz XTAL clock.
3	Reserved	RES	Reserved for future use.
2:0	PriRecClkSel (88E6321 only)	RWS to 0x7	Synchronous Ethernet Primary Recovered Clock Select. This field indicates the internal PHY number whose recovered clock will be presented on the GPIO pin which is configured to select SE_RCLK0. The reset value of 0x7 selects no clock for SE_RCLK0. The SE_RCLK0 frequency is 25 MHz when PriRecClkSel points to a physical PHY or SERDES (3 for PHY 3, etc). Selecting a port where there is no PHY (or SERDES) will connect the SE_RCLK0 signal to the internal 25 MHz XTAL clock.

Table 270: TAI Global Config Register, AVBPort = 0xE
Offset: 0x9 or Decimal 9

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



Link Street®

88E6321/88E6320 Functional Specification

Table 271: TAI Global Config Register, AVBPort = 0xE
Offset: 0xA or Decimal 10

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

Table 272: TAI Global Config Register, AVBPort = 0xE
Offset: 0xB or Decimal 11

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

Table 273: TAI Global Config Register, AVBPort = 0xE
Offset: 0xE or Decimal 14

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

Table 274: TAI Global Config Register, AVBPort = 0xE
Offset: 0xF or Decimal 15

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

Table 275: TAI Global Config Register, AVBPort = 0xE
Offset: 0x10 or Decimal 16

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



Link Street®

88E6321/88E6320 Functional Specification

Table 276: TAI Global Config Register, AVBPort = 0xE
Offset: 0x11 or Decimal 17

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

Table 277: TAI Global Config Register, AVBPort = 0xE
Offset: 0x12 or Decimal 18

Bits	Field	Type	Description
15:4	Reserved	RES	Reserved for future use.
3:0	Lock Correction	RO	Trig Lock Correction amount. When the TrigLock bit is set to a one (TAI offset 0x00) enabling a potential clock adjustment, these bits are cleared to zero. When the TrigLock bit is cleared to zero (indicating that the requested clock adjustment is now past in time) these bits will reflect the magnitude and direction that was applied to the PTP_TRIG leading edge of the generated clock. A value of zero means no adjustment was necessary. If bit 3 is a one then the leading edge of the clock was moved n number of clocks earlier in time where n is shown in bits 2:0. If bit 3 is a zero then the leading edge of the clock was moved n number of clocks later in time where n is shown in bits 2:0.

Table 278: TAI Global Config Register, AVBPort = 0xE
Offset: 0x13 or Decimal 19

Bits	Field	Type	Description
15:0	TrigGenDelay	RWR	Trigger Generation Delay. This field specifies the number of PTP clocks the PTP_TRIG Delayed waveform is delayed from the PTP_TRIG waveform. A value of 0x0000 is zero clocks delay and if the Delay is greater than the PTP_TRIG cycle time then the delay is considered zero. This delay time can optional be used by the Qav Shapers as a Deblocking Window (see the TimeAwareDeBlock bit in Qav Port offsets 0x00 to 0x06).

Table 279: TAI Global Config Register, AVBPort = 0xE
Offset: 0x14 or Decimal 20

Bits	Field	Type	Description
15:0	TrigGen2Time[15:0]y	RWR	<p>Trigger Generation 2 Time bits 15 to 0 of a 32-bit register. This field specifies the PTP Global Time (TAI offsets 0x0E & 0x0F) where the 1st leading edge of PTP_TRIG 2 will occur (with a three TSCLKPer latency) when PTP Trig is in the continuous square wave mode (i.e, when TrigMode is 0x0, offset 0x00 above) as long as this register's value is non-zero. If its value is zero, the 1st leading edge of PTP_TRIG 2 will occur when TrigGenReg is set to a one (TAI offset 0x00).</p> <p>This register is also used to for re-locking the leading edge of the square wave (see TrigLock in TAI offset 0x00).</p> <p>The upper 16-bits of this register are contained in the register below.</p>

Table 280: TAI Global Config Register, AVBPort = 0xE
Offset: 0x15 or Decimal 21

Bits	Field	Type	Description
15:0	TrigGen2Time[31:16]y	RWR	<p>Trigger Generation 2 Time bits 31:16 of a 32-bit register. See the description above.</p> <p>The lower 16-bits of this register are contained in the register above..</p>

Table 281: TAI Global Config Register, AVBPort = 0xE
Offset: 0x16 or Decimal 22

Bits	Field	Type	Description
15:4	Reserved	RES	Reserved for future use.
3:0	Lock2Correction	RO	<p>Trig Lock 2 Correction amount. When the TrigLock bit is set to a one (TAI offset 0x00) enabling a potential clock adjustment, these bits are cleared to zero. When the TrigLock bit is cleared to zero (indicating that the requested clock adjustment is now past in time) these bits will reflect the magnitude and direction that was applied to the PTP_TRIG 2 leading edge of the generated clock.</p> <p>A value of zero means no adjustment was necessary.</p> <p>If bit 3 is a one then the leading edge of the clock was moved n number of clocks earlier in time where n is shown in bits 2:0. If bit 3 is a zero then the leading edge of the clock was moved n number of clocks later in time where n is shown in bits 2:0.</p>



Link Street®

88E6321/88E6320 Functional Specification

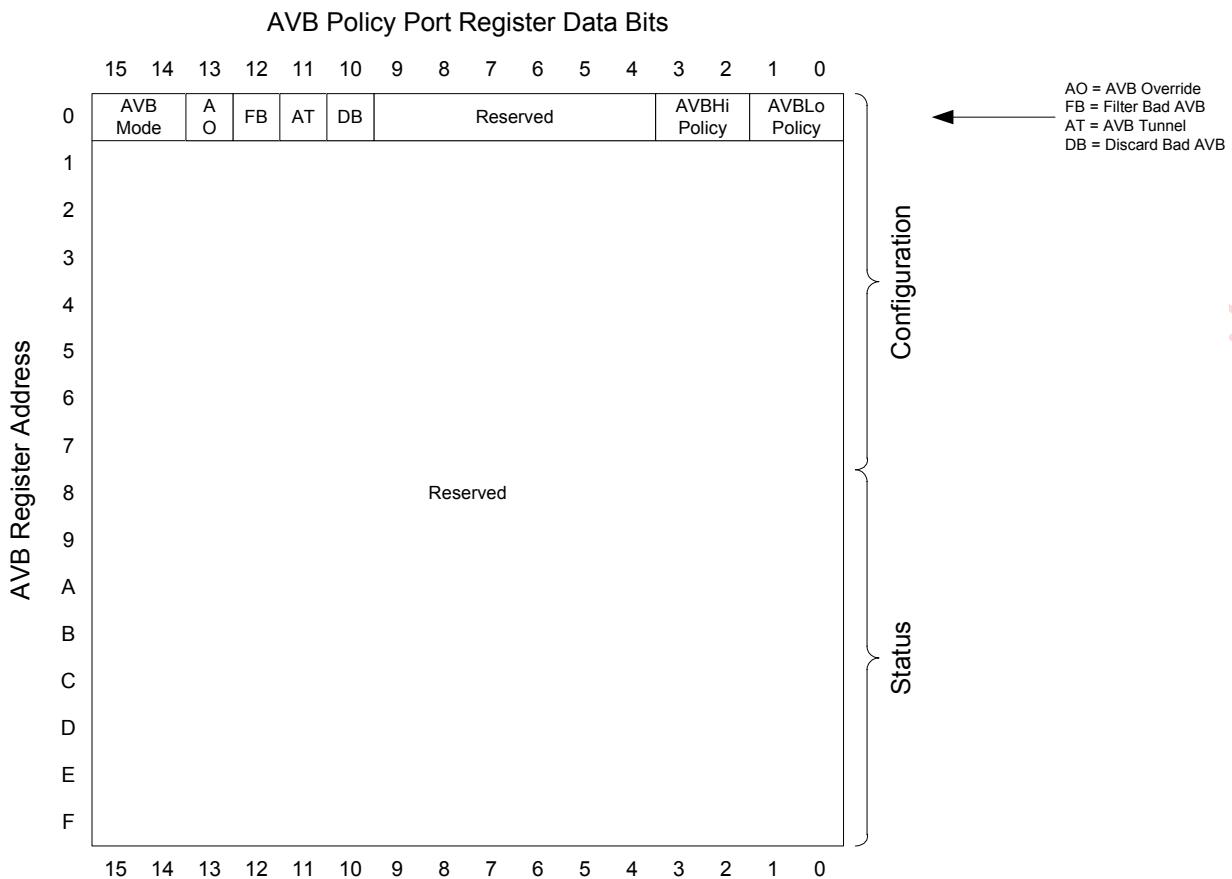
Table 282: TAI Global Config Register, AVBPort = 0xE
Offset: 0x17 or Decimal 23

Bits	Field	Type	Description
15:0	TrigGen2Delay	RWR	Trigger Generation 2 Delay. This field specifies the number of PTP clocks the PTP_TRIG Delayed 2 waveform is delayed from the PTP_TRIG 2 waveform. A value of 0x0000 is zero clocks delay and if the Delay is greater than the PTP_TRIG 2 cycle time then the delay is considered zero. This delay time can optional be used by the Qav Shapers as a Blocking Window (see the TimeAwareBlock bit in Qav Port offsets 0x01 to 0x07).

10.6.2 AVB Registers

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

Figure 70: AVB Policy Port Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x1 & AVBPort = 6:0)





Link Street®

88E6321/88E6320 Functional Specification

Table 283: AVB Policy Register

Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15:14	AvbMode	RWR	<p>Port's AVB Mode. These bits are used to select the AVB mode on the port as follows:</p> <ul style="list-style-type: none">0 = Legacy port mode. All frames entering this port are considered Legacy unless they are overridden by the frame's DA¹ in which case they are considered AVB.1 = Standard AVB port mode. Any tagged frame (Provider or 802.1Q tagged) that ends up with² an AVB frame priority³ is considered AVB. All other frames are considered Legacy.2 = Enhanced AVB port mode. Any frame that ends up with an AVB frame priority whose DA is contained in the ATU with an AVB Entry State is considered AVB. All other frames are considered Legacy. Frames that end up with an AVB frame priority but whose DA is not in ATU with an AVB Entry State are considered Bad AVB frames (which can be filtered – see Filter Bad Avb below).3 = Secure AVB port mode. Any frame that ends up with an AVB frame priority whose DA is contained in the ATU with an AVB Entry State and whose DPV has this source port's bit set to a one, is considered AVB. All other frames are considered Legacy. Frames that end up with an AVB frame priority but whose DA is not in ATU with an AVB Entry State or whose DPV does not have this source port's bit set to a one, are considered Bad AVB frames (which can be filtered – see Filter Bad Avb below). <p>AVB frames are allowed to use the AVB queues on other AVB egress ports (those ports where AvbMode <> 0x0 – see AvbOverride below). Legacy frames cannot use AVB queues.</p>
13	AvbOverride	RWR	AVB Override. When this bit is cleared to a zero, normal frame processing occurs. When this bit is set to a one the egress portion of this port is considered AVB even if the ingress portion is not (because AvbMode, above, is set to 0x0). This allows this port's egress to support AVB queues and flows even though the port is a Legacy port. This can be useful if an AVB flow is allowed to egress the AVB cloud.
12	FilterBadAvb	RWR	Filter Bad AVB frames. When this bit is cleared to a zero, normal frame processing occurs. When this bit is set to a one, frames that are considered Bad AVB frames (see AvbModes 2 & 3 above) are filtered using the method determined by the DiscardBadAvb bit below . This can be useful to prevent AVB steams from flowing before the path is completely set up.
11	AvbTunnel	RWR	<p>AVB Tunnel. When this bit is cleared to a zero normal frame processing occurs. When this bit is set to a one the port based VLANTable masking, 802.1Q VLAN membership masking and the Trunk Masking is bypassed for any frame entering this port that is considered AVB by DA. This includes unicast as well as multicast frames.</p> <p>A frame is considered AVB by DA if its DA is in the ATU with an AVB Entry State with priority override where the overridden priority equals the Hi or Lo AVB frame priorities (AVB Policy Global offset 0x0) and where the port's DA Priority Override bits are 0x1 (bits 15:14 of this register).</p> <p>NOTE: Do not set this bit to a 0x1 if the port's VLANTunnel bit (Port offset Register 0x04) is set to a 0x1.</p>

Table 283: AVB Policy Register (Continued)

Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
10	DiscardBadAvb	RWR	<p>Discard Bad AVB frames. When the FilterBadAvb bit, above, is set to a one this bit determines the type of filtering that will occur on frames that are considered Bad AVB frames (see AvbModes 2 & 3 above).</p> <p>When this bit is cleared to a zero, Bad AVB frames are prevented from egressing out AVB ports only. In this mode Bad AVB frames are still allowed to egress non-AVB ports, however. This keeps the Bad AVB frames out of the AVB queues but allows them to egress non-AVB ports which may be best for the network.</p> <p>When this bit is set to a one, Bad AVB frames are prevented from egressing out all ports.</p> <p>NOTE: An AVB egress port is one whose AVB Mode is \neq 0x0 or whose AVB Override bit is set to a one (see bits above).</p>
9:4	Reserved	RES	Reserved for future use.
3:2	Avb Hi Policy	RWR	<p>Avb Hi Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then AVB Hi Policy Mapping can occur on this port for AVB Hi frames. AVB Hi Policy Mapping occurs when the DA of a frame is contained in the ATU address database (Global offset 0x0C) with an Entry State that indicates AVB with priority override where the overridden priority equals the Hi AVB frame priority (AVB Policy Global offset 0x0) and when the port's DA Priority Override bits are 0x1 (bits 15:14 of this register). When this occurs the mapping of nonfiltered frames is determined by the setting of these bits as follows:</p> <ul style="list-style-type: none"> 00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Reserved <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor.</p> <p>Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>



Link Street®

88E6321/88E6320 Functional Specification

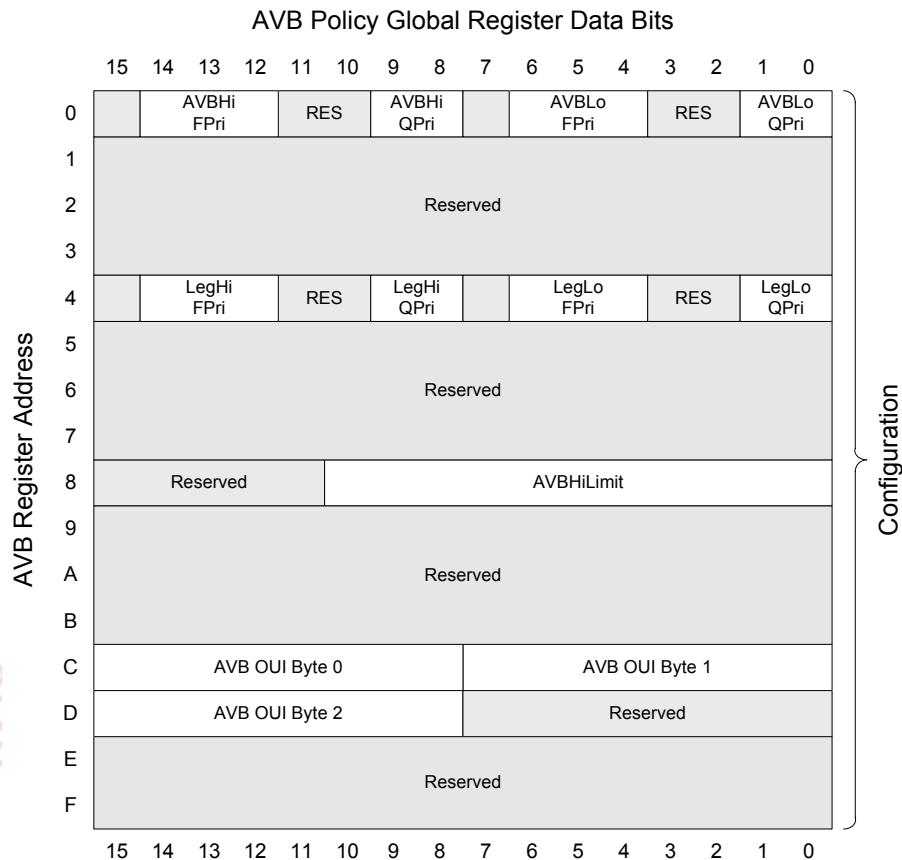
Table 283: AVB Policy Register (Continued)

Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
1:0	Avb Lo Policy	RWR	<p>Avb Lo Policy Mapping. When these bits are cleared to a zero normal frame switching occurs. When either or both of these bits are set to a one then AVB Lo Policy Mapping can occur on this port for AVB Lo frames. AVB Lo Policy Mapping occurs when the DA of a frame is contained in the ATU address database (Global offset 0x0C) with an Entry State that indicates AVB with priority override where the overridden priority equals the Lo AVB frame priority (AVB Policy Global offset 0x0) and when the port's DA Priority Override bits are 0x1 (bits 15:14 of this register). When this occurs the mapping of nonfiltered frames is determined by the setting of these bits as follows:</p> <p>00 = Normal frame switching 01 = Mirror (copy) frame to the MirrorDest port (global offset 0x1A) 10 = Trap (re-direct) frame to the CPUDest port (global offset 0x1A) 11 = Reserved</p> <p>Mirrored frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Monitor. Trapped frames that egress a DSA or EtherType DSA port (port offset 0x04) will egress as a To_CPU frame with a Code of Policy Trap.</p>

1. An AVB DA override requires the frame's DA to be in the ATU with an AVB Entry State with priority override and an AVB Hi or AVB Lo FPri in the ATU entry and the port's DA Priority Override mode (Port offset 0x0D) is set to 0x1.
2. Frame priority, or FPri, can be modified by many mechanisms inside the switch.
3. An AVB frame priority is an FPri that is equal to AvbHiFPri or AvbLoFPri (AVB Policy Global offset 0x00).

Figure 71: AVB Policy Global Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x1 & AVBPort = 0xF





Link Street®

88E6321/88E6320 Functional Specification

Table 284: AVB Policy Global Clock Register, AVBPort = 0xF
Offset: 0x0 or Decimal 0

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

Table 285: AVB Policy Global Legacy Register, AVBPort = 0xF
Offset: 0x4 or Decimal 4

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

Table 286: AVB Policy Global Limit Register, AVBPort = 0xF
Offset: 0x8 or Decimal 8

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



Link Street®

88E6321/88E6320 Functional Specification

Table 287: AVB Policy Global OUI Register Byte 0 & 1, AVBPort = 0xF
Offset: 0xC or Decimal 12

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

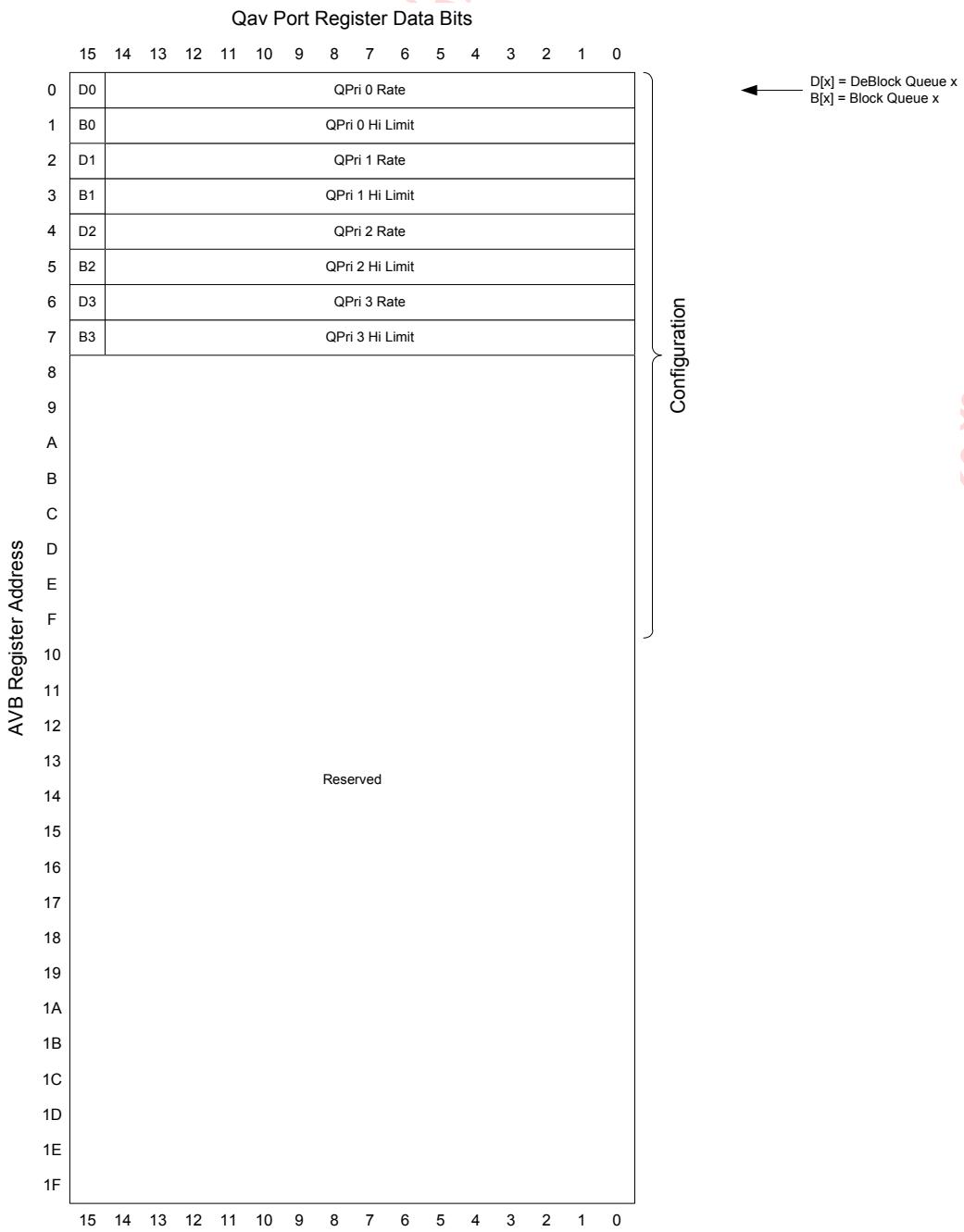
Table 288: AVB Policy Global OUI Register Byte 2, AVBPort = 0xF
Offset: 0xD or Decimal 13

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

10.6.3 Qav Registers

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

Figure 72: Qav Port Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x2 & AVBPort = 6:0)





Link Street®

88E6321/88E6320 Functional Specification

Table 289: QavPort Config Register
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15	TimeAwareDeBlock	RWR	Time Aware Qav Deblocking enable. When this bit is cleared to zero normal Qav operation occurs. When this bit is set to a one, this queue will Deblock (i.e., allow to transmit) frames present in this queue during the PTP_TRIG / Delay Trig window as configured by TrigGenDelay (TAI offset 0x13). Deblock means that frames become eligible for transmission and connecting this function to the PTP TAI allows this event to be synchronized to the gPTP Grand Master clock. Otherwise when this bit is set to a one, frames present in this queue will not be eligible for transmission until the next PTP_TRIG / Delay Trig Deblocking window.
14:0	QPri 0 Rate	RWR	Priority Queue 0 Rate. A value of 0x000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as: $\text{QPri 0 Rate} = \text{Desired Rate} / 32k \text{ bits per sec}$ For example, for a Desired Rate of 64 kbps set this register to a value of 0x002 (64 kbps / 32kbps = 2). NOTE: The minimum supported rate is 32 kbps and the rate increments supported are 32 kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).

Table 290: QavPort Config Register
Offset: 0x1 or Decimal 1

Bits	Field	Type	Description
15	TimeAwareBlock	RWR	Time Aware Qav Blocking enable. When this bit is cleared to zero normal Qav operation occurs. When this bit is set to a one, this queue will Block (i.e., prevent from transmitting) all frames present in this queue during the PTP_TRIG 2 / Delay 2 Trig window as configured by TrigGen2Delay (TAI offset 0x17). Blocking means that frames present in this queue will not become eligible for transmission and connecting this function to the PTP TAI allows this event to be synchronized to the gPTP Grand Master clock.
14:0	QPri 0 BurstBytes HiLimit	RWS 0x7FFF	Priority Queue 0 Hi Limit This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for this queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 32,768 bytes is enabled on this queue.

Table 291: QavPort Config Register
Offset: 0x2 or Decimal 2

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:0	QPri 1 Rate	RWR	<p>Priority Queue 1 Rate. A value of 0x000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as:</p> <p>QPri 0 Rate = Desired Rate / 32k bits per sec</p> <p>For example, for a Desired Rate of 64 kbps set this register to a value of 0x002 (64 kbps/ 32kbps = 2).</p> <p>NOTE: The minimum supported rate is 32 kbps and the rate increments supported are 32 kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).</p>

Table 292: QavPort Config Register
Offset: 0x3 or Decimal 3

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:0	QPri 1 HiLimit	RWS 0x7FFF	<p>Priority Queue 1 Hi Limit</p> <p>This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for this queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach).</p> <p>By default the maximum credit of 32,768 bytes is enabled on this queue.</p>

Table 293: QavPort Config Register
Offset: 0x4 or Decimal 4

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:0	QPri 2 Rate	RWR	<p>Priority Queue 2 Rate. A value of 0x000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as:</p> <p>QPri 0 Rate = Desired Rate / 32k bits per sec</p> <p>For example, for a Desired Rate of 64 kbps set this register to a value of 0x002 (64 kbps/ 32kbps = 2).</p> <p>NOTE: The minimum supported rate is 32 kbps and the rate increments supported are 32 kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 294: QavPort Config Register
Offset: 0x5 or Decimal 5

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:0	QPri 2 HiLimit	RWS 0x7FFF	Priority Queue 2 Hi Limit This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for the queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 32,768 bytes is enabled on this queue.

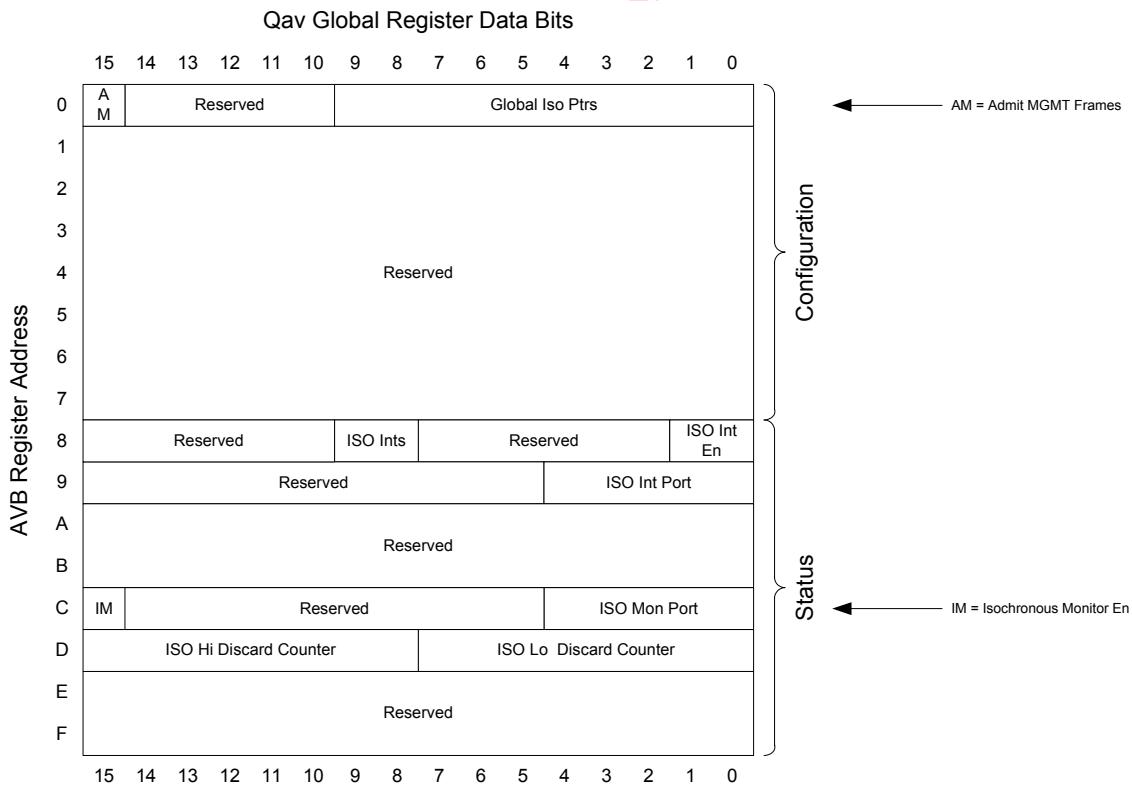
Table 295: QavPort Config Register
Offset: 0x6 or Decimal 6

Bits	Field	Type	Description
15:11	Reserved	RES	Reserved for future use.
10:0	QPri3 Rate	RWR	Priority Queue 3 Rate. A value of 0x000 disables this queue's Qav rate shaper allowing data in this queue to flow as if there were no shaper. This register is used to specify the information rate for this queue priority. The rate in bits per sec is calculated as: QPri 0 Rate = Desired Rate / 32k bits per sec For example, for a Desired Rate of 64 kbps set this register to a value of 0x002 (64 kbps/ 32kbps = 2). NOTE: The minimum supported rate is 32 kbps and the rate increments supported are 32 kbps with a maximum supported rate of 1000 Mbps (i.e., the speed of the port).

Table 296: QavPort Config Register
Offset: 0x7 or Decimal 7

Bits	Field	Type	Description
15	Reserved	RES	Reserved for future use.
14:0	QPri3 HiLimit	RWS 0x7FFF	Priority Queue 3 Hi Limit This value specifies the number of credits in bytes that can be accumulated when the queue is blocked from sending out a frame. This limit specifies the maximum subsequent burst size allowed for the queue once the queue becomes unblocked (i.e., this register determines the maximum value Qav's hiCredit is allowed to reach). By default the maximum credit of 32,768 bytes is enabled on this queue.

Figure 73: QavGlobal Register bit Map (Global 2 offset 0x16 & 0x17 w/AVBBlock = 0x2 & AVBPort = 0xF)





Link Street®

88E6321/88E6320 Functional Specification

Table 297: Qav Global Config Register, AVBPort = 0xF
Offset: 0x0 or Decimal 0

Bits	Field	Type	Description
15	Admt MGMT	RWR	<p>Admit Management Frames. This bit only has an affect when the management frames are sharing one of the isochronous queues. When set to 0x1, this bit allows the queue controller to always accept management frames and the packet memory pointer is taken from the isochronous pointer pool. In case of queue congestion, AVB frames are dropped first and management frames are always accepted. When set to 0x0, in congestion situations, the queue controller may drop the management frames based on AVB congestion policy.</p> <p>NOTE: This bit is expected to be set only in fully managed environments as a management storm could have an adverse affect on traffic through multiple ports.</p>
14:10	Reserved	RES	Reserved for future use.
9:0	Global IsoPtrs	RWR	<p>Global Isochronous Queue Pointer Threshold</p> <p>This field indicates the total number of isochronous pointers that are reserved for isochronous streams. The value is expected to be computed in SRP software and programmed into hardware based on the total aggregate isochronous streams configured to go through this device.</p>

Table 298: Qav Global Status Register, AVBPort = 0xF
Offset: 0x8 or Decimal 8

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

Table 299: Qav Global Status Register, AVBPort = 0xF
Offset: 0x9 or Decimal 9

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



Link Street®

88E6321/88E6320 Functional Specification

Table 300: Qav Global Status Register, AVBPort = 0xF
Offset: 0xC or Decimal 12

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

Table 301: Qav Global Status Register, AVBPort = 0xF
Offset: 0xD or Decimal 13

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

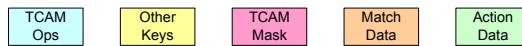
10.6.4

Switch Global 3 Registers - TCAM (88E6321) Only

The device contains a third set of global registers that effect all the Ethernet ports in the device. Each Global 3 register is 16-bits wide and their bit assignments are shown in [Figure 74](#), [Figure 75](#), and [Figure 76](#).

Figure 74: Global 3 Register bit Map for TCAM Page 0 (Device Addr 0x1D)

Global Register Data Bits																																							
0	TB	TCAMOp	0	0	RES	TCAM Entry																																	
1	Reserved																																						
2	Mask for Type				Frame Type	RES																																	
3	Mask for SPV				SPV																																		
4	Mask for Res or PPRI/PVID				Reserved or PPRI+PVID[11:8]																																		
5	Mask for Index or PVID				Index or PVID[7:0]																																		
6	Mask for bits 7:0 of this register				Frame Octet 1 (DA1) or Octet 49																																		
7	Mask for bits 7:0 of this register				Frame Octet 2 (DA2) or Octet 50																																		
8	Mask for bits 7:0 of this register				Frame Octet 3 (DA3) or Octet 51																																		
9	Mask for bits 7:0 of this register				Frame Octet 4 (DA4) or Octet 52																																		
A	Mask for bits 7:0 of this register				Frame Octet 5 (DA5) or Octet 53																																		
B	Mask for bits 7:0 of this register				Frame Octet 6 (DA6) or Octet 54																																		
C	Mask for bits 7:0 of this register				Frame Octet 7 (SA1) or Octet 55																																		
D	Mask for bits 7:0 of this register				Frame Octet 8 (SA2) or Octet 56																																		
E	Mask for bits 7:0 of this register				Frame Octet 9 (SA3) or Octet 57																																		
F	Mask for bits 7:0 of this register				Frame Octet 10 (SA4) or Octet 58																																		
10	Mask for bits 7:0 of this register				Frame Octet 11 (SA5) or Octet 59																																		
11	Mask for bits 7:0 of this register				Frame Octet 12 (SA6) or Octet 60																																		
12	Mask for bits 7:0 of this register				Frame Octet 13 (Tag1) or Octet 61																																		
13	Mask for bits 7:0 of this register				Frame Octet 14 (Tag2) or Octet 62																																		
14	Mask for bits 7:0 of this register				Frame Octet 15 (PRI+VID) or Octet 63																																		
15	Mask for bits 7:0 of this register				Frame Octet 16 (VID) or Octet 64																																		
16	Mask for bits 7:0 of this register				Frame Octet 17 (Type1) or Octet 65																																		
17	Mask for bits 7:0 of this register				Frame Octet 18 (Type2) or Octet 66																																		
18	Mask for bits 7:0 of this register				Frame Octet 19 (Data1) or Octet 67																																		
19	Mask for bits 7:0 of this register				Frame Octet 20 (Data2) or Octet 68																																		
1A	Mask for bits 7:0 of this register				Frame Octet 21 (Data3) or Octet 69																																		
1B	Mask for bits 7:0 of this register				Frame Octet 22 (Data4) or Octet 70																																		
1C	Reserved																																						
1D	Reserved																																						
1E	Reserved																																						
1F	Reserved																																						





Link Street®

88E6321/88E6320 Functional Specification

10.6.5

Common Global 3 Registers for all TCAM Pages:

The Global 3 registers are used to control the Ternary Content Addressable Memory (TCAM). This large memory is accessed as three separate Global 3 pages. Global 3 offset 0x00 is common to each of the pages while offset 0x01 to 0x1F will be different per page. This section covers the common register at offset 0x00.

Table 302: TCAM Operation Register

Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	TCAMBusy	SC	TCAM Busy. This bit must be set to a one to start a TCAM operation (see TCAMOp below). Only one TCAM operation can be executing at one time so this bit must be zero before setting it to a one. When the requested TCAM operation completes this bit will automatically be cleared to a zero.
14:12	TCAMOp	RWR	TCAM Opcode. The device supports the following TCAM operations (all of these operations can be executed while frames are transiting through the switch): 000 = No Operation 001 = Flush ¹ All entries 010 = Flush ² or invalidate a single TCAM entry 011 = Load ³ an entry 100 = Get Next ⁴ (read next valid entry – all pages) 101 = Read ⁵ an entry's page (performs a direct read of an entry) 110 = Reserved 111 = Reserved
11:10	TCAM Page	RWR	TCAM Page. Each TCAM Entry is made up of 3 pages of data. All three pages need to loaded in a particular order for the TCAM to operate correctly while frames are flowing through the switch. If the entry is currently valid, it must first be Flushed ⁶ . Then Page 2 needs to be loaded first, followed by Page 1, and then finally Page 0. Each page load requires its own Write TCAMOp with these TCAM Page bits set accordingly. When Page 0 is loaded the TCAM will immediately take affect on the next frame that enters the switch (assuming the TCAM is enabled). When two TCAM entries are being used for 96 byte frame compares ⁷ , the TCAM entry for bytes 49 to 96 needs to loaded first before the TCAM entry for bytes 1 to 48 is entered. The same Page 2, Page 1 then Page 0 process must be used. If these two entries are currently valid, they both must be Flushed prior to the loading of the new data. The TCAM Page bits are also used to address the desired TCAM Page to be loading into Global 2 offsets 0x02 through 0x01B for the Read TCAMOp. NOTE: These bits must be 0x0 for the Get Next, Flush All and Flush an entry TCAMOps.
9:8	Reserved	RES	Reserved for future use.
7:0	TCM Entry	RWR	TCAM Entry. These bits must point to the desired TCAM entry for Flush an entry (010), Load an entry (011) and Read an entry (101) TCAMOps. These bits return the TCAM entry found for the Get Next (100) TCAMOp.

1. A Flush All command will initialize TCAM Pages 0 and 1, offsets 0x02 to 0x1B to 0x0000, and TCAM Page 2 offset 0x02 to 0x05 to 0x0000 and TCAM Page 2 offset 0x02 to 0x05 to 0x0000 for all TCAM entries with the exception that TCAM Page 0 offset 0x02 will be initialized to 0x0FF.

2. A Flush a single TCAM entry command will write the same values to a TCAM entry as a Flush All command, but it is done to the selected single TCAM entry only.

3. The Load sequence of a TCAM entry is critical. See the text for more information.

4. A Get Next operation finds the next higher TCAM Entry number that is valid (i.e., any entry whose Page 0 offset 0x02 is not equal to 0x0FF). The TCAM Entry register (bits 7:0) is used as the TCAM entry to start from. To find the lowest number TCAM Entry that is valid, start the Get Next operation with TCAM Entry set to 0xFF. When the operation is done the TCAM Entry register contains the next higher valid TCAM entry number and Global 3 offsets 0x02 to 0x1B registers contain the data found in that TCAM's entry for the TCAM Page 0. To read back the other pages of this TCAM entry, use the Read TCAMOp. To find the next higher entry simply issue the Get Next opcode again. If the TCAM Entry register is returned set to all one's and its TCAM Page 0 offset 0x02 is equal to 0x0FF, no higher valid TCAM entry was found (and the end of the table was reached). If the TCAM Entry register is returned set to all one's and its TCAM Page 0 offset 0x02 is not equal to 0x0FF, the highest TCAM entry was found as valid, and the end of the table was reached.
5. A Read TCAMOp loads the Global 3 offsets 0x02 to 0x1B registers with the data found in the TCAM entry and its TCAM page pointed to by the TCAM Entry and TCAM Page bits of this register (bits 7:0 and 11:10 respectively).
6. Use the Flush an entry TCAM entry TCAMOp to do this (bits 14:12 of this register).
7. 96 byte compares will result in a TCAM hit only if the ingressing frame is at least 96 bytes in size.



Link Street®

88E6321/88E6320 Functional Specification

10.6.6

Global 3 Registers for TCAM Page 0:

TCAM Key Data and Frame Match Data bytes are in the lower 8 bits of each 16-bit register. The upper 8 bits of each register are the Mask bits for the lower 8 bits where bit 15 is the mask for bit 7, bit 14 is the mask for bit 6, etc. The individual pairs of data bits and mask bits work together as follows:

Mask	Data	Meaning
0	0	Don't Care. The data bit can be a one or a zero for a TCAM hit to occur.
1	0	Hit on 0. The data bit must be a zero for a TCAM hit to occur.
1	1	Hit on 1. The data bit must be a one for a TCAM hit to occur.
0	1	Never Hit. Used to prevent a TCAM hit from occurring from this entry ¹ .

The Never Hit value is used to Flush the TCAM or Purge a TCAM entry. On a TCAM Flush or Purge, this value is written to the 1st TCAM byte only (offset 0x02 on TCAM page 1). On a TCAM Flush All or on a TCAM Flush an entry, all other TCAM data and mask bytes are written to a value of 0x0000 and so are the Action bytes.

1. This is needed so that a TCAM entry can be defined as unused or invalid.

Table 303: Keys Register 1
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description										
15:14	Mask	RWR	Mask for bits 7:6 below. Both of these bits must be zeros for unused entries and must be ones for valid entries.										
13:8	Reserved	RWR	Reserved for future use. These bits must always be zeros.										
7:6	Frame Type	RW	<p>Frame Type. These bits are used to define the Frame type or mode this TCAM entry is defined for as follows:</p> <p>For bytes 1 to 48 of a TCAM Entry:</p> <ul style="list-style-type: none"> 00 = Normal Network frame, 0x8100 tagged or untagged 01 = DSA Tagged¹ (this selection changes the definition of the tag bytes – TCAM Page 1 offsets 0x12 to 0x15) 10 = Provider Tagged² (this selection changes the definition of TCAM Page 1 offsets 0x04 and 0x05) 11 = Reserved for future use (except to indicate this TCAM entry is unused if bits 15:14 are zero). Do not use this setting if bits 15:14 are ones. <p>For bytes 49 to 96 (or beyond) of a TCAM Entry:</p> <ul style="list-style-type: none"> 00 = Continued Entry³ for bytes 49 to 96 (or beyond) 01 = Reserved 10 = Reserved 11 = Reserved <p>NOTE: DSA Tagged mode should only be used on TCAM entries that are associated to ports that are in DSA or Ether Type DSA mode (see FrameMode in Port offset 0x04). Likewise Provider Tagged mode should only be used on TCAM entries that are associated to ports that are in Provider mode. It can be useful for Normal Network mode to be associated to a Provider port – that way non-Provider tagged frames that enter that Provider port can be processed by that TCAM entry. Normal Network mode entries are also useful on Ether Type DSA tag mode ports.</p> <p>NOTE: These TCAM Entry Frame Type bits are matched to the resulting mode each frame is determined to be as it is being received. This is directly related to the Port's Frame Mode setting (Port offset 0x04) as follows:</p> <table border="0"> <tr> <td>Port's Frame Mode</td> <td>Possible Resulting Frame Types on 1st 48 Byte TCAM compares</td> </tr> <tr> <td>00 or Normal</td> <td>00 or Normal</td> </tr> <tr> <td>01 or DSA</td> <td>01 or DSA</td> </tr> <tr> <td>10 or Provide</td> <td>10 or Provider (if Provider Tagged) else 00 or Normal</td> </tr> <tr> <td>11 or DSA</td> <td>01 or DSA (if Ether Type DSA) else 00 or Normal</td> </tr> </table>	Port's Frame Mode	Possible Resulting Frame Types on 1st 48 Byte TCAM compares	00 or Normal	00 or Normal	01 or DSA	01 or DSA	10 or Provide	10 or Provider (if Provider Tagged) else 00 or Normal	11 or DSA	01 or DSA (if Ether Type DSA) else 00 or Normal
Port's Frame Mode	Possible Resulting Frame Types on 1st 48 Byte TCAM compares												
00 or Normal	00 or Normal												
01 or DSA	01 or DSA												
10 or Provide	10 or Provider (if Provider Tagged) else 00 or Normal												
11 or DSA	01 or DSA (if Ether Type DSA) else 00 or Normal												
5:0	Reserved but special - read description	RWS	Reserved for future use. These bits must be ones for unused entries and must be zeros for valid entries.										

1. Ethertype DSA tagged frames get their Ether Type and following 2 null bytes removed during ingress. Therefore the rest of the TCAM match data must match the standard DSA frame format.
2. Provider Tagged frames get their Provider Tag (or multiple Provider Tags if there are more than one) removed during ingress. Therefore the rest of the TCAM match data must match the modified frame without the Provider Tag(s). The 1st Provider Tag that was removed can have its tag data matched in the TCAM – see TCAM Page 1 offsets 0x04 and 0x05.
3. A Continued Entry has a non-zero Index value – see Page 0 offset 0x05.



Link Street®

88E6321/88E6320 Functional Specification

Table 304: Keys Register 2

Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15	Reserved	RWR	Reserved for future use.
14:8	Mask	RWR	Mask for bits 7:0 of this register. All valid TCAM entries must have its SPV mask bits set to a one for the ports that are specifically not to be associated with this entry, see the SPV bit description below.
7	Reserved	RWR	Reserved for future use.
6:0	SPV	RWR	<p>Source Port Vector. These bits are used to define which switch ports can use this TCAM entry. This way one TCAM entry can be associated with more than one port if those ports are to take the same Action for the same kind of frame. Bit 0 is associated with Port 0, bit 1 with Port 1, etc. Use the SPV and Mask bits as follows:</p> <p>If a TCAM entry is to be associated with only one port then set all the SPV Mask bits to a one and set the SPV to the bit vector of the port. For example: If an entry is to be associated with Port 5 only, then set the SPV Mask bits above to 0x7F and the SPV bits in this register to 0x20.</p> <p>Alternatively the method below can be used for a single port as well.</p> <p>If a TCAM entry is to be associated with more than one port but specifically excluded from other ports, then set all the excluded port's SPV Mask bits above to one and set the SPV bits in this register to all zeros. For example: If an entry is to be associated with Port 2 and Port 5 only, then set the SPV Mask bits to 0x5B and the SPV bits to 0x00. What is really being done here is specifically excluding specific ports from using this entry as their bits must be zero for this entry to match.</p> <p>If a TCAM entry is to be associated with all the ports in the device then set the SPV bits to a don't care value by setting the SPV Mask bits and the SPV bits in this register to 0x00.</p> <p>NOTE: A port must be enabled to use the TCAM (Port offset 0x0D) or that port's bits in this register will have no effect.</p> <p>These bits are matched to a vectorized conversion of the ingressing frame's physical source port number.</p>

Table 305: Keys Register 3
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15:8	Mask	RWR	Mask bits for 7:0 of this register.
7:4	Reserved or PPRI	RWR	Reserved for future use when the TCAM entry's FrameMode bits (TCAM Page 0, offset 0x02) are not equal to Provider Tagged or if this is a TCAM entry for searches beyond the first 48 bytes. When the TCAM entry's FrameMode bits are Provider Tagged, these bits are Provider Priority bits. When a port is in Provider Mode (see FrameMode, Port offset 0x04), and if the ingress frame is provider tagged, the provider tag(s) are removed from the frame and the 1st provider tag's data is saved. That saved data is compared to these bits. Bits 7:5 are the frame's Provider PRI bits and bit 4 is the frame's Provider DE (Drop Eligible) bit.
3:0	Reserved or PVID[11:8]	RWR	Reserved for future use when the TCAM entry's FrameMode bits (TCAM Page 0, offset 0x02) are not equal to Provider Tagged or if this is a TCAM entry for searches beyond the first 48 bytes. When the TCAM entry's FrameMode bits are Provider Tagged, these bits are Provider VID[11:8] bits. When a port is in Provider Mode (see FrameMode, Port offset 0x04), and if the ingress frame is provider tagged, the provider tag(s) are removed from the frame and the 1st provider tag's data is saved. That saved data is compared to these bits. Bits 3:0 are the frame's Provider VID[11:8] bits.



Link Street®

88E6321/88E6320 Functional Specification

Table 306: Keys Register 4

Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15:8	Mask	RWR	Mask bits 7:0 of this register.
7:0	Index or PVID[7:0]	RWR	<p>When the TCAM entry's FrameMode bits are Provider Tagged, these bits are Provider VID[7:0] bits for TCAM entries for the first 48 bytes of a frame only. When a port is in Provider Mode (see FrameMode, Port offset 0x04), and if the ingressing frame is provider tagged, the provider tag(s) are removed from the frame and the 1st provider tag's data is saved. That saved data is compared to these bits. Bits 7:0 are the frame's Provider VID[7:0] bits.</p> <p>When the TCAM entry's FrameMode bits are not equal to Provider Tagged or if this is a TCAM entry for TCAM searches beyond the first 48 bytes, these bits are the Index bits. Index is used to concatenate two TCAM entries to form a 96 byte (or larger) frame data lookup. If a TCAM entry only needs to look at the first 48 bytes of a frame, then these bits must be 0x00 (with a Mask of 0xFF) or these bits can be set to don't care (i.e., clear this register to 0x0000). If a TCAM entry needs to support a 96 byte (or larger) lookup, two (three or more) TCAM entries are required. Load the TCAM match data for bytes 49-96 first with a unique¹, non-zero number in this Index field. A recommended number to use is that entry's TCAM Entry number (TCAM offset 0x00) as that will be a unique number. Then load the TCAM match data for bytes 1 to 48 with this field cleared to 0x00, set that entry's Continue Action bit to a one (TCAM page 2 offset 0x02), and place the value of the upper match entry's Index field into the lower match entry's Next Index field (TCAM page 2 offset 0x03).</p>

1. The TCAM entry that is matching the frame's upper data (bytes 49-96) must have a non-zero Index value. That ensures it will not become a hit when the 1st 48 bytes of the frame are being matched (the 1st 48 byte matches of a frame always assert a 0x00 value to compare with this Index field). If the 1st 48 byte compare returns a 'hit' that needs to be extended to frame bytes 49 to 96 (or beyond), it signals this by setting its Continue Action register bit to a one (TCAM Page 2 offset 0x02). The connection between the 1st 48 byte match data and the 2nd (3rd or 4th) 48 bytes is done through the Next Index data. The 1st 48 byte compare is the 1st part of this connection. It must return a non-zero value in its Next Index data (TCAM Page 2 offset 0x03). That Next Index data is then used as the data to match to the Index field for the upper 48 byte TCAM entry or entries when comparing frame data bytes 49 to 96 (or beyond). If a 'hit' occurs in that 2nd TCAM lookup, the actions from that 2nd 'hit' are used for the frame. If there is no 'hit' for bytes 49 to 96 (or beyond), then there is no 'hit' for the frame (even though there might have been a 'hit' for the 1st 48 bytes of the frame). The Index value for 96 bytes compares don't have to be unique if there more than one upper TCAM entries that would be valid as possible 'hits' to the same lower 48 byte match. As this is hard to think about, it is recommended that a unique value be used for the Index and that its value be set to the value of the upper entries TCAM Entry value (i.e., where it is actually stored in the TCAM).

Table 307: Match Data Register 1
Offset: 0x06 or Decimal 6

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 1 or 49	RWR	Frame Octet 1 or 49. This is the match data for octet 1 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 49.

Table 308: Match Data Register 2
Offset: 0x07 or Decimal 7

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 2 or 50	RWR	Frame Octet 2 or 50. This is the match data for octet 2 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 50.

Match Data Register 3 through 22 continue on in the same way supporting match data for frame byte 3 (or 51) up to frame byte 22 (or 70) (or 118). These registers are found in TCAM Page 0 offsets 0x08 to 0x1B. All match bytes work the same way.

The Tag match bytes (frame bytes 13 to 16 or TCAM Page 0 offsets 0x12 to 0x15) need to be configured as follows based on the desired affect:

- For TCAM entries that can ‘hit’ on Normal Network frames that are either 802.1Q Tagged or Untagged: Set the entry’s FrameMode (TCAM Page 0, offset 0x02) to 00 and set all four Tag match bytes (TCAM Page 0, offsets 0x12 to 0x15) to 0x0000 (don’t care). The TCAM compare logic takes Untagged frames and shifts them up by four bytes starting after the frame’s Source Address field so the same TCAM entry can match both Tagged and Untagged frames.
- For TCAM entries that can ‘hit’ on Normal Network frames that are Untagged only: Set the entry’s FrameMode (TCAM Page 0, offset 0x02) to 00 and set all four Tag match bytes (TCAM Page 0, offsets 0x12 to 0x15) to 0xFF00 (match zeros). The TCAM compare logic takes Untagged frames and shifts them up by four bytes starting after the frame’s Source Address field and inserts four bytes of zero. Tagged frames won’t look this way.
- For TCAM entries that can ‘hit’ on Normal Network frames that are Tagged only: Set the entry’s FrameMode (TCAM Page 0, offset 0x02) to 00 and set the upper two Tag match bytes (TCAM Page 0, offsets 0x12 to 0x13) to 0xFF81 and 0xFF00 respectively (match 0x8100). Only Tagged frames look this way. Set the lower two Tag match bytes (TCAM Page 0, offset 0x14 to 0x15) to don’t care (0x00FF) if the value of the Tag is important. Otherwise the TCAM entry can be set to match part or all of Tag’s data bits.
- For TCAM entries that can ‘hit’ on Provider Tagged only: Set the entry’s FrameMode (TCAM Page 0, offset 0x02) to 10. The Tag match bytes (TCAM Page 0, offset 0x12 to 0x15) are not used to match to the Provider Tag data as they are used to match a Customer tag that might exist in the frame (as defined above). If the TCAM entry needs to match to parts of the Provider Tag data, this data can be matched using Key register 3 and 4 (TCAM Page 0, offset 0x04 and 0x05).
- For TCAM entries that can ‘hit’ on DSA Tagged only: Set the entry’s FrameMode (TCAM Page 0, offset 0x02) to 01. The four Tag match bytes (TCAM Page 0, offset 0x12 to 0x15) are used to match to the DSA Tag data where enabled.

10.6.7 Global 3 Registers for TCAM Page 1

Figure 75: Global 3 Register bit Map for TCAM Page 1 (Device Addr 0x1D)

Global Register Data Bits																									
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
0	TB	TCAMOp	0	1	RES												0								
TCAM Entry																									
1	Reserved																								
2	Mask for bits 7:0 of this register							Frame Octet 23 (Data7) or Octet 71																	
3	Mask for bits 7:0 of this register							Frame Octet 24 (Data7) or Octet 72																	
4	Mask for bits 7:0 of this register							Frame Octet 25 (Data7) or Octet 73																	
5	Mask for bits 7:0 of this register							Frame Octet 26 (Data8) or Octet 74																	
6	Mask for bits 7:0 of this register							Frame Octet 27 (Data9) or Octet 75																	
7	Mask for bits 7:0 of this register							Frame Octet 28 (Data10) or Octet 76																	
8	Mask for bits 7:0 of this register							Frame Octet 29 (Data11) or Octet 77																	
9	Mask for bits 7:0 of this register							Frame Octet 30 (Data12) or Octet 78																	
A	Mask for bits 7:0 of this register							Frame Octet 31 (Data13) or Octet 79																	
B	Mask for bits 7:0 of this register							Frame Octet 32 (Data14) or Octet 80																	
C	Mask for bits 7:0 of this register							Frame Octet 33 (Data15) or Octet 81																	
D	Mask for bits 7:0 of this register							Frame Octet 34 (Data16) or Octet 82																	
E	Mask for bits 7:0 of this register							Frame Octet 35 (Data17) or Octet 83																	
F	Mask for bits 7:0 of this register							Frame Octet 36 (Data18) or Octet 84																	
10	Mask for bits 7:0 of this register							Frame Octet 37 (Data19) or Octet 85																	
11	Mask for bits 7:0 of this register							Frame Octet 38 (Data20) or Octet 86																	
12	Mask for bits 7:0 of this register							Frame Octet 39 (Data21) or Octet 87																	
13	Mask for bits 7:0 of this register							Frame Octet 40 (Data22) or Octet 88																	
14	Mask for bits 7:0 of this register							Frame Octet 41 (Data23) or Octet 89																	
15	Mask for bits 7:0 of this register							Frame Octet 42 (Data24) or Octet 90																	
16	Mask for bits 7:0 of this register							Frame Octet 43 (Data25) or Octet 91																	
17	Mask for bits 7:0 of this register							Frame Octet 44 (Data26) or Octet 92																	
18	Mask for bits 7:0 of this register							Frame Octet 45 (Data27) or Octet 93																	
19	Mask for bits 7:0 of this register							Frame Octet 46 (Data28) or Octet 94																	
1A	Mask for bits 7:0 of this register							Frame Octet 47 (Data29) or Octet 95																	
1B	Mask for bits 7:0 of this register							Frame Octet 48 (Data30) or Octet 96																	
1C	Reserved																								
1D	Reserved																								
1E	Reserved																								
1F	Reserved																								

TCAM Ops Other Keys TCAM Mask Match Data Action Data

Table 309: TCAM Operation Register
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	TCAMBusy	SC	TCAM Busy. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
14:12	TCAMOp	RWR	TCAM Opcode. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
11:10	TCAM Page	RWR	TCAM Page. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
9:8	Reserved	RES	Reserved for future use.
7:0	TCM Entry	RWR	TCAM Entry. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.

Table 310: Match Data Register 23
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 23 or 71	RWR	Frame Octet 23 or 71. This is the match data for octet 23 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 71.

Table 311: Match Data Register 24
Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:8	Mask	RWR	Mask for bits 7:0 of this register.
7:0	Frame Octet 24 or 72	RWR	Frame Octet 24 or 72. This is the match data for octet 24 of the frame if this TCAM entry is for the first 48 bytes of a frame. If this TCAM entry is for the second 48 bytes of a frame this is the match data for octet 72.

Match Data Register 25 through 48 continue on in the same way supporting match data for frame byte 24 (or 73) up to frame byte 48 (or 96). These registers are found in TCAM Page 1 offsets 0x04 to 0x1B. All match bytes work the same way.

10.6.8 Global 3 Registers for TCAM Page 2:

Figure 76: Global 3 Register bit Map for TCAM Page 2 (Device Addr 0x1D)

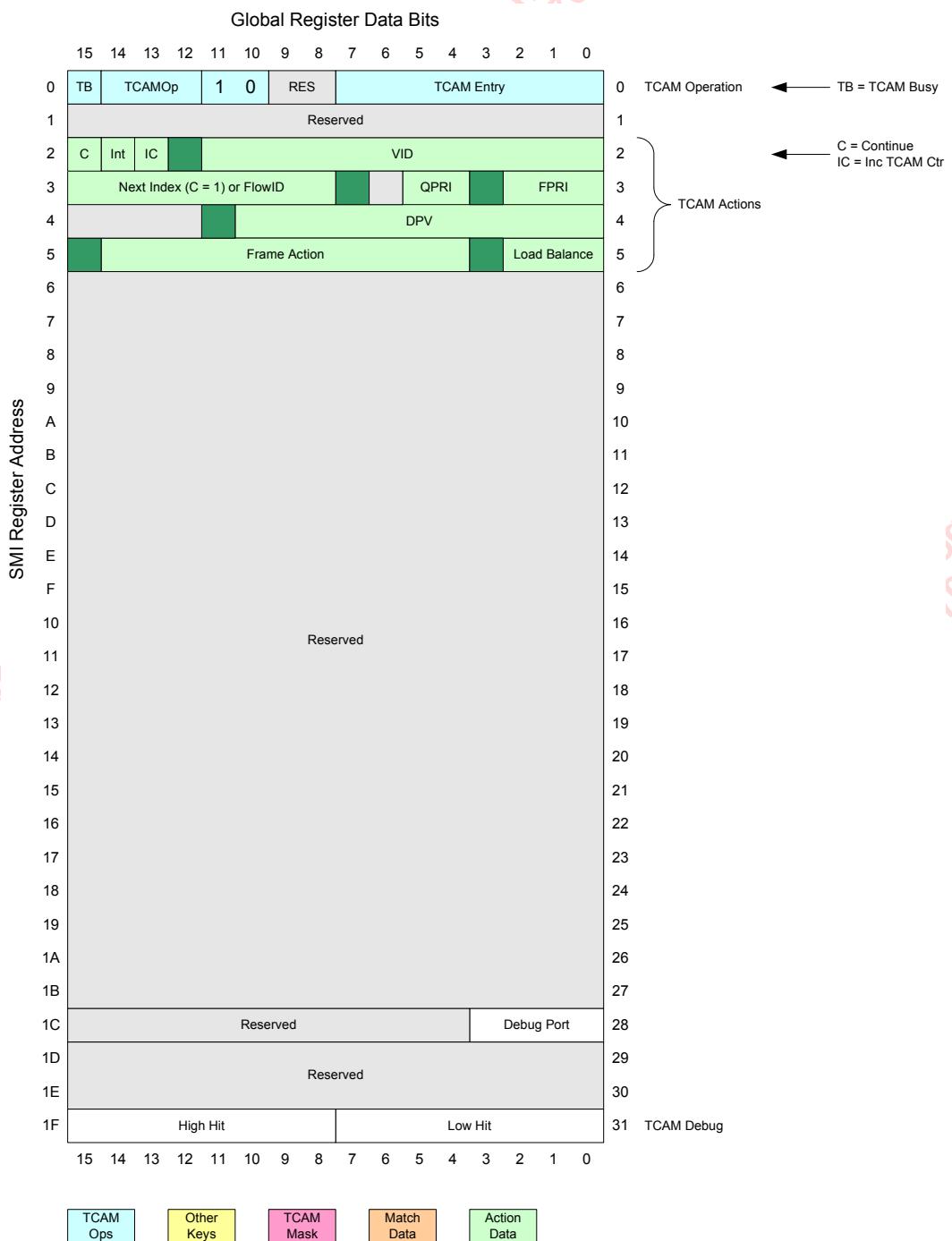


Table 312: TCAM Operation Register
Offset: 0x00 or Decimal 0

Bits	Field	Type	Description
15	TCAMBusy	SC	TCAM Busy. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
14:12	TCAMOp	RWR	TCAM Opcode. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
11:10	TCAM Page	RWR	TCAM Page. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.
9:8	Reserved	RES	Reserved for future use.
7:0	TCAM Entry	RWR	TCAM Entry. This register is the same across all TCAM pages. Please see this register's full description under TCAM Page 0.

Table 313: Action Register 1
Offset: 0x02 or Decimal 2

Bits	Field	Type	Description
15	Continue	RWR	Continue this TCAM entry. If this TCAM entry only needs to support a compare in the first 48 bytes of a frame, or if this is the TCAM entry covers bytes 49 to 96 of a frame, set this bit to a zero. This bit should only be a 1 on TCAM entries that cover the first 48 bytes of a frame that needs to be extended to also match bytes 49 to 96 of the frame. See the Next Index bits in Action Register 2, offset 0x03.
14	Int	RWR	Interrupt on a TCAM hit. When this bit is set to a one on a TCAM entry (where the Continue bit is a zero), a TCAM hit interrupt will be generated whenever a match occurs to this entry. The results of the interrupt are stored in the TCAM Int bit (Global 1, offset 0x00).
13	IncTcamCtr	RWR	Increment the port's TCAM Counter pointed to by FlowID[7:6] on a TCAM hit. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the port where the frame ingressed that generated this TCAM hit will get one of the port's TcamCtr[3:0] counters (Global 1 offset 0x1D) incremented by 1. The counter that gets incremented is determined by the TCAM entry's FlowID[7:6] bits (TCAM Page 2, offset 0x03). If FlowID[7:6] = 0 the port's TcamCtr[0] will be incremented, if FlowID[7:6] = 1 the port's TcamCtr[1] will be incremented, etc..
12	VID Override	RWR	VID Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the VID Data (bits 11:0 below) are assigned to the frame overriding any other VID assignment in the switch.
11:0	VID Data	RWR	VID Override Data. When the VID Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this VID Data is assigned to the frame. This VID data is used for ATU Lookups and Learns and it is the VID that will appear in the frame if the frame egresses tagged.



Link Street®

88E6321/88E6320 Functional Specification

Table 314: Action Register 2

Offset: 0x03 or Decimal 3

Bits	Field	Type	Description
15:8	Next Index or Flow ID	RWR	<p>Next Index or Flow ID. When the Continue bit (TCAM Page 2, offset 0x02) is a one, it means that this TCAM 'hit' is not done yet. A subsequent TCAM search needs to be done with the next 48 bytes of the frame (up to byte 96 and beyond). To distinguish the subsequent search from the first search, the subsequent search is done with this Next Index data being used as the next search's Index key (TCAM Page 0, offset 0x05).</p> <p>When the Continue bit (TCAM Page 2, offset 0x02) is a zero, these bits are the Flow ID. FlowID is used as follows:</p> <p>Flow ID[1:0] is decoded and sent to the Port Ingress Rate Limiter (PIRL – Global 2 offsets 0x09 & 0x0A) such that up to 4 flows per port can be limited and/or counted separately as long a FlowID[5] = 0 (see below).</p> <p>FlowID[4:0] is placed into the Egress Header when Header Type = 0x0 (Global 1 offset 0x1C). This way the receiving device, typically a CPU, can use these bits as a Flow indicator. FlowID[4:0] will be zeros in the Egress Header on all TCAM misses.</p> <p>FlowID[5] = 0 validates FlowID[1:0] to PIRL. If FlowID[5] = 1 then FlowID[1:0] are ignored by PIRL.</p> <p>FlowID[7:6] is used to indicate which of the ingress port's four TcamCtr's to increment when the IncTcamCtr bit = 1 (TCAM Page 2, offset 0x02).</p>
7	QPRI Override	RWR	QPRI Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the QPRI Data (bits 5:4 below) are assigned to the frame overriding any other QPRI assignment in the switch.
6	Reserved	RES	Reserved for future use.
5:4	QPRI Data	RWR	QPRI Override Data. When the QPRI Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this QPRI Data is assigned to the frame. This QPRI data is used to determine which internal switch queue the frame should be mapped into.
3	FPRI Override	RWR	FPRI Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the FPRI Data (bits 2:0 below) are assigned to the frame overriding any other FPRI assignment in the switch.
2:0	FPRI Data	RWR	FPRI Override Data. When the FPRI Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this FPRI Data is assigned to the frame. This FPRI data is the PRI that will appear in the frame if the frame egresses tagged.

Table 315: Action Register 3
Offset: 0x04 or Decimal 4

Bits	Field	Type	Description
15:12	Reserved	RES	Reserved for future use.
11	DPV Override	RWR	DPV Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the DPV Data (bits 6:0 below) are assigned to the frame overriding any other DPV assignment in the switch.
10:7	Reserved	RES	Reserved for future use.
6:0	DPV Data	RWR	DPV Override Data. When the DPV Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this DPV Data is assigned to the frame. This DPV data is used to map which port or ports a frame egresses. Set bit 0 to a one to have a frame egress Port 0. Set bit 1 to a one to a frame egress Port 1, etc. If the DPV Data that is assigned to the frame is all zeros, the frame will be Filtered and the Port's InFiltered counter will increment accordingly ¹ . In this case, to prevent this filtered frame from being mirrored, this TCAM entry must also set the FAction Override to a one, with the FAction Data bits all zeros (Global 3 Page 2 offset 0x05). NOTE: Source Port filtering will not be done.

1. The Port's InFiltered counter will increment only if Port offset 0x12 counters are configured to be InFiltered counters – see Port offset 0x12.



Link Street®

88E6321/88E6320 Functional Specification

Table 316: Action Register 4

Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
15	FAction Override	RWR	Frame Action Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the Frame Action Data (bits 14:4 below) are assigned to the frame overriding any other Frame Action assignment in the switch.
14:4	FAction Data	RWR	<p>Frame Action Override Data. When the Frame Action Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this Frame Action Data is assigned to the frame. The Frame Action bits are assigned as follows:</p> <p>14 – Source is Tagged 13 – ProviderVID 12 – MGMT 11 – ARP 10 – Snoop 9 – PolMirror 8 – PolTrap 7 – SaNRL 6 – DaNRL 5 – Reserved 4 – CFI</p> <p>The above Frame Action data is assigned to the frame at the end of the Ingress Processing. This updated data is then passed to the Port Ingress Rate Limiter (PIRL) and the Queue Controller (QC). PIRL can be programmed to look at MGMT, ARP, PolMirror, PolTrap, SaNRL and DaNRL bits and in the process limit how many of these frames are allowed to ingress a port or let some frames bypass their normal limiting. The QC also uses these bits to modify the frame's processing as follows:</p> <ul style="list-style-type: none">• Source is Tagged (Src_Tag) is used to indicate if an ingressing frame is Tagged (Src_Tag = 1) or not. The egress logic uses this information to know how to modify the frame when it is directed to transmit the frame Tagged or UnTagged. DSA egress ports also use this information to add the DSA tag or to convert the frame's existing tag to a DSA tag. Forcing Src_Tag to zero will cause a tag (DSA or normal) to be inserted even if the ingressing frame was tagged.• ProviderVID causes the inserted Provider Tag (on Provider Tagged egress ports) to get the frame's VID from the assignment made in Ingress instead of using the source Port's DefaultVID (Port offset 0x07). This bit must be set to a 1 if this TCAM entry is assigning a new VID for frames that egress out a Provider Port – or that new VID will not be in the Provider's Tag.• MGMT allows frames to egress Blocked ports (Port offset 0x04) and prevents the use of tagging information from the VTU (i.e., the frame will egress unmodified unless the port is a Header or DSA port). It sets the MGMT bit in the egress Marvell Header. If the frame egresses a DSA port the frame will be marked as a To_CPU frame with a MGMT CPU Code.• ARP only has a effect if ARP mirroring is enabled on the ingress port (Port offset 0x08). If that is the case, it will cause this frame to be copied to the port indicated by CPU Dest (Global 1, offset 0x1A). If the CPU Dest port's FrameMode is a DSA mode (Port offset 0x04) the copied frame will egress with a To_CPU tag with an APR CPU Code. ARP will become a Trap (instead of a Mirror) if the DPV of the frame is zeros.

Table 316: Action Register 4 (Continued)

Offset: 0x05 or Decimal 5

Bits	Field	Type	Description
14:4 (cont.)	FAction Data (cont.)	RWR	<ul style="list-style-type: none"> Snoop allows frames to egress Blocked ports (Port offset 0x04) and prevents the use of tagging information from the VTU (i.e., the frame will egress unmodified unless the port is a Header or DSA port). It sets the Snoop bit in the egress Marvell Header. If the frame egresses a DSA port the frame will be marked as a To_CPU frame with a IGMP CPU Code. PolMirror it will cause this frame to be copied to the port indicated by Mirror Dest (Global 1, offset 0x1A). If the Mirror Dest port's FrameMode is a DSA mode (Port offset 0x04) the copied frame will egress with a To_CPU tag with a PolMirror CPU Code. PolMirror will become a Trap (instead of a Mirror) if the DPV of the frame is zeros. PolTrap allows frames to egress Blocked ports (Port offset 0x04) and prevents the use of tagging information from the VTU (i.e., the frame will egress unmodified unless the port is a Header or DSA port). It sets the PolTap bit in the egress Marvell Header (assuming the SpinnakerAv Header updates are done) . If the frame egresses a DSA port the frame will be marked as a To_CPU frame with a PolTrap CPU Code. SaNRL and DaNRL do not modify the frame's content in any way but do override the normal source of these signals to PIRL. PIRL can then be programmed to use these signals as an indication of what frames can bypass PIRL's normal frame rate limiting (see PRIL Bucket Configuration, offset 0x07). CFI defines the CFI bit in tagged customer frames or the DE bit in Provider Tagged frames. <p>NOTE: All of these Frame Actions are sent to PIRL where they can be used to limit this frame or used to prevent its limiting.</p>
3	Load Balance Override	RWR	Load Balance Override Enable. When this bit is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), the Load Balance Data (bits 2:0 below) are assigned to the frame overriding any other Load Balance assignment in the switch.
2:0	Load Balance Data	RWR	Load Balance Override Data. When the Load Balance Override bit, above, is set to a one on a TCAM entry that is hit (where the Continue bit is a zero), this Load Balance Override Data is assigned to the frame. This Load Balance data is used to map which port or ports a frame egresses. This data is used to access the Trunk Masking Table (Global 2, offset 0x07) for this frame instead of using the results from the frame's DA & SA. NOTE: Source Port filtering will be done.



Link Street®

88E6321/88E6320 Functional Specification

Table 317: TCAM Debug Register

Offset: 0x01C or Decimal 28

Bits	Field	Type	Description
15:4	Reserved	RES	Reserved for future use.
3:0	Debug Port	RWR	Debug Port number. This register is used to define the port number whose TCAM activities will be tracked in the TCAM Debug registers in offset 0x1F below.

Table 318: TCAM Debug Register

Offset: 0x01F or Decimal 31

Bits	Field	Type	Description
15:8	High Hit	RO	TCAM Entry for High 48-byte Hit. This register gets updated with the TCAM Entry number that was a 'hit' for any compare to frame bytes 49 to 96 for the port number defined in offset 0x1C above. This register will get overwritten with every new 'hit' for frame bytes 49 to 96 and it gets written to zeros on every Low Hit update.
7:0	Low Hit	RO	TCAM Entry for Low 48-byte Hit. This register gets updated with the TCAM Entry number that was a 'hit' for any compare to frame bytes 1 to 48 for the port number defined in offset 0x1C above. This register will get overwritten with every new 'hit' for frame bytes 1 to 48. It gets written to ones at reset.

The above registers work as follows in the following example sequence:

- 0x00FF = Reset state, no hits yet
- 0x0000 = 48-byte TCAM hit on Entry 0x00
- 0x0201 = 96-byte TCAM hit on Entries 0x01 (low) and 0x02 (high)
- 0x0003 = 48-byte TCAM hit on Entry 0x03
- 0x0809 = 96-byte TCAM hit on Entries 0x09 (low) and 0x08 (high)
- 0x0009 = 48-byte TCAM hit on Entry 0x09 with a TCAM miss on the high 48-bytes
- 0x00FF = Miss on 1st 48 byte lookup

**Note**

A TCAM 'hit' on comparisons to frame bytes 1 to 48 are 'partial hits' if the selected TCAM entry (the lowest entry that matched) has the Continue bit set to a one. But these 'partial hits' are still recorded in the lower 8 bits of the TCAM Debug Register (above) so it can aide in debugging.

**Note**

Final TCAM 'hits' occur only if the last TCAM comparision results in a 'hit' to an entry whose Continue bits is cleared to a zero. Only this last entry's Action bits will be used on the frame.

11

PHY and SERDES Register Description



Note

The PHY and SERDES registers in the device are not accessible directly. Instead, these registers must be accessed through the SMI PHY Command and Data Switch Registers in the Global 2 register space (Global 2 Offsets 0x18 and 0x19).

11.1

PHY Register Description

The device supports both Clause 22 MDIO register access protocol and Clause 45 XMDIO register access protocol. The device also supports Clause 22 MDIO access to registers in Clause 45 XMDIO space using Page 0 register 13 and 14.

Table 319 below defines the register types used in the register map.

Table 319: Register Types

Type	Description
LH	Register field with latching high function. If status is high, then the register is set to a one and remains set until a read operation is performed through the management interface or a reset occurs.
LL	Register field with latching low function. If status is low, then the register is cleared to zero and remains zero until a read operation is performed through the management interface or a reset occurs.
Retain	Value written to the register field does not take effect without a software reset, and the register maintains its value after a software reset.
RES	Reserved for future use. All reserved bits are read as zero unless otherwise noted.
RO	Read only.
ROC	Read only clear. After read, register field is cleared to zero.
R/W	Read and write with initial value indicated.
RWC	Read/Write clear on read. All bits are readable and writable. After reset or after the register field is read, register field is cleared to zero.
SC	Self-Clear. Writing a one to this register causes the desired function to be immediately executed, then the register field is automatically cleared to zero when the function is complete.
Update	Value written to the register field does not take effect until soft reset is executed; however, the written value can be read even before the software reset.
WO	Write only. Reads to this type of register field return undefined data.

For all binary equations appearing in the register map, the symbol | is equal to a binary OR operation.



Link Street®

88E6321/88E6320 Functional Specification

11.1.1

MDIO Register Description

The IEEE defines only 32 registers address space for the PHY. In order to extend the number of registers address space available a paging mechanism is used. For register address 0 to 21, and 23 to 28 register 22 bits 7 to 0 are used to specify the page. There is no paging for register 22.

In this document, the short hand used to specify the registers take the form register_page.bit:bit, register_page.bit, register.bit:bit, or register.bit.

For example:

Register 16 page 2 bits 5 to 2 is specified as 16_2.5:2.

Register 16 page 2 bits 5 is specified as 16_2.5.

Register 2 bit 3 to 0 is specified as 2.3:0.

Note that in this context the setting of the page register (register 22) has no effect.

Register 2 bit 3 is specified as 2.3.

Table 320: Device Register Map Summary - Page 0 - Page 7

Register Address	Page Address							
	0	1	2	3	4	5	6	7
	Copper		MAC Ctrl/Status	LED		Advanced VCT	Pakct Gen/Chk	Cable Diagnostics
0	Copper Control Register							
1	Copper Status Register							
2	PHY Identifier 1							
3	PHY Identifier 2							
4	Copper Auto-Negotiation Advertisement Register							
5	Copper Link Partner Ability Register - Base Page							
6	Copper Auto-Negotiation Expansion Register							
7	Copper Next Page Transmit Register							
8	Copper Link Partner Next Page Register							
9	1000BASE-T Control Register							
10	1000BASE-T Status Register							
11								
12								
13	MMD access control register							
14	MMD access Address/Data register							
15	Extended Status Register							
16	Copper Specific Control Register 1		MAC Specific Control Register 1	LED[3:0] Function Control Register		Advanced VCT TX to MDI[0] Rx Coupling	Packet Generation	PHY Cable Diagnostics Pair 0 Length
17	Copper Specific Status Register 1			LED[3:0] Polarity Control Register		Advanced VCT TX to MDI[1] Rx Coupling	CRC Counters	PHY Cable Diagnostics Pair 1 Length
18	Copper Specific Interrupt Enable Register		MAC Specific Interrupt Enable Register	LED Timer Control Register		Advanced VCT TX to MDI[2] Rx Coupling	Checker Control	PHY Cable Diagnostics Pair 2 Length
19	Copper Interrupt Status Register		MAC Specific Status Register	LED[5:4] Function Control and Polarity Register		Advanced VCT TX to MDI[3] Rx Coupling		PHY Cable Diagnostics Pair 3 Length
20	Copper Specific Control Register 2		Copper RX_ER Byte Capture			1000BASE-T Pair Skew Register		PHY Cable Diagnostics Results
21	Copper Specific Receive Error Counter		MAC Specific Control Register 2			1000BASE-T Pair Swap and Polarity		PHY Cable Diagnostics Control
22	Page Address							
23	Global Interrupt Status					Advance VCT Control		
24						Advanced VCT Sample Point Distance		
25						Advanced VCT Cross Pair Positive Threshold		Advanced VCT Cross Pair Negative Threshold
26	Copper Specific Control Register 3					Advanced VCT Same Pair Impedance Positive Threshold 0 and 1	Misc Test	Advanced VCT Same Pair Impedance Negative Threshold 0 and 1
27						Advanced VCT Same Pair Impedance Positive Threshold 2 and 3	Temperature Sensor	Advanced VCT Same Pair Impedance Negative Threshold 2 and 3
28						Advanced VCT Same Pair Impedance Positive Threshold 4 and Transmit Pulse Control		Advanced VCT Same Pair Impedance Negative Threshold 4
29								
30								
31								



Link Street®

88E6321/88E6320 Functional Specification

Table 321: PHY Register Map

Register Name	Register Address	Table and Page
Copper Control Register	Page 0, Register 0	Table 322, p. 448
Copper Status Register	Page 0, Register 1	Table 323, p. 450
PHY Identifier 1	Page 0, Register 2	Table 324, p. 451
PHY Identifier 2	Page 0, Register 3	Table 325, p. 451
Copper Auto-Negotiation Advertisement Register	Page 0, Register 4	Table 326, p. 452
Copper Link Partner Ability Register - Base Page	Page 0, Register 5	Table 327, p. 455
Copper Auto-Negotiation Expansion Register	Page 0, Register 6	Table 328, p. 456
Copper Next Page Transmit Register	Page 0, Register 7	Table 329, p. 456
Copper Link Partner Next Page Register	Page 0, Register 8	Table 330, p. 457
1000BASE-T Control Register	Page 0, Register 9	Table 331, p. 457
1000BASE-T Status Register	Page 0, Register 10	Table 332, p. 458
XMDIO MMD Access Control Register	Page 0, Register 13	Table 333, p. 459
XMDIO MMD Access Address/Data Register	Page 0, Register 14	Table 334, p. 459
Extended Status Register	Page 0, Register 15	Table 335, p. 460
Copper Specific Control Register 1	Page 0, Register 16	Table 336, p. 460
Copper Specific Status Register 1	Page 0, Register 17	Table 337, p. 462
Copper Specific Interrupt Enable Register	Page 0, Register 18	Table 338, p. 464
Copper Interrupt Status Register	Page 0, Register 19	Table 339, p. 465
Copper Specific Control Register 2	Page 0, Register 20	Table 340, p. 466
Copper Specific Receive Error Counter Register	Page 0, Register 21	Table 341, p. 466
Page Address	Page Any, Register 22	Table 342, p. 467
Global Interrupt Status	Page 0, Register 23	Table 343, p. 467
Copper Specific Control Register 3	Page 0, Register 26	Table 344, p. 468
MAC Specific Control Register 1	Page 2, Register 16	Table 345, p. 470
MAC Specific Interrupt Enable Register	Page 2, Register 18	Table 346, p. 471
MAC Specific Status Register	Page 2, Register 19	Table 347, p. 472
Copper RX_ER Byte Capture	Page 2, Register 20	Table 348, p. 472
MAC Specific Control Register 2	Page 2, Register 21	Table 349, p. 473
Advanced VCT™ TX to MDI[0] Rx Coupling	Page 5, Register 16	Table 350, p. 474
Advanced VCT™ TX to MDI[1] Rx Coupling	Page 5, Register 17	Table 351, p. 475
Advanced VCT™ TX to MDI[2] Rx Coupling	Page 5, Register 18	Table 352, p. 476
Advanced VCT™ TX to MDI[3] Rx Coupling	Page 5, Register 19	Table 353, p. 477
1000BASE-T Pair Skew Register	Page 5, Register 20	Table 354, p. 478
1000BASE-T Pair Swap and Polarity	Page 5, Register 21	Table 355, p. 478
Advanced VCT™ Control	Page 5, Register 23	Table 356, p. 479
Advanced VCT™ Sample Point Distance	Page 5, Register 24	Table 357, p. 479
Advanced VCT™ Cross Pair Positive Threshold	Page 5, Register 25	Table 358, p. 480

Table 321: PHY Register Map (Continued)

Register Name	Register Address	Table and Page
Advanced VCT Same Pair Impedance Positive Threshold 0 and 1	Page 5, Register 26	Table 359, p. 480
Advanced VCT™ Same Pair Impedance Positive Threshold 2 and 3	Page 5, Register 27	Table 360, p. 480
Advanced VCT™ Same Pair Impedance Positive Threshold 4 and Transmit Pulse Control	Page 5, Register 28	Table 361, p. 481
Copper Port Packet Generation	Page 6, Register 16	Table 362, p. 481
Copper Port CRC Counters	Page 6, Register 17	Table 363, p. 482
Checker Control	Page 6, Register 18	Table 364, p. 482
Misc Test	Page 6, Register 26	Table 365, p. 483
Temperature Sensor	Page 6, Register 27	Table 366, p. 483
PHY Cable Diagnostics Pair 0 Length	Page 7, Register 16	Table 367, p. 484
PHY Cable Diagnostics Pair 1 Length	Page 7, Register 17	Table 368, p. 484
PHY Cable Diagnostics Pair 2 Length	Page 7, Register 18	Table 369, p. 484
PHY Cable Diagnostics Pair 3 Length	Page 7, Register 19	Table 370, p. 484
PHY Cable Diagnostics Results	Page 7, Register 20	Table 371, p. 485
PHY Cable Diagnostics Control	Page 7, Register 21	Table 372, p. 486
Advanced VCT™ Cross Pair Negative Threshold	Page 7, Register 25	Table 373, p. 486
Advanced VCT Same Pair Impedance Negative Threshold 0 and 1	Page 7, Register 26	Table 374, p. 486
Advanced VCT Same Pair Impedance Negative Threshold 2 and 3	Page 7, Register 27	Table 375, p. 487
Advanced VCT Same Pair Impedance Negative Threshold 4	Page 7, Register 28	Table 376, p. 487



Link Street®

88E6321/88E6320 Functional Specification

Table 322: Copper Control Register
Page 0, Register 0

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Copper Reset	R/W, SC	0x0	SC	Copper Software Reset. Affects pages 0, 2, 3, 5, and 7. Writing a 1 to this bit causes the PHY state machines to be reset. When the reset operation is done, this bit is cleared to 0 automatically. The reset occurs immediately. 1 = PHY reset 0 = Normal operation
14	Loopback	R/W	0x0	0x0	When loopback is activated, the transmitter data presented on TXD is looped back to RXD internally. Link is broken when loopback is enabled. Loopback speed is determined by Registers 21_2:0. 1 = Enable Loopback 0 = Disable Loopback
13	Speed Select (LSB)	R/W	0x0	Update	Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect. A write to this register bit does not take effect until any one of the following also occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down Bit 6, 13 11 = Reserved 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps
12	Auto-Negotiation Enable	R/W	0x1	Update	Changes to this bit are disruptive to the normal operation. A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down If Register 0_0.12 is set to 0 and speed is manually forced to 1000 Mbps in Registers 0.13 and 0.6, then Auto-Negotiation will still be enabled and only 1000BASE-T full-duplex is advertised if register 0_0.8 is set to 1, and 1000BASE-T half-duplex is advertised if 0.8 is set to 0. Registers 4.8:5 and 9.9:8 are ignored. Auto-Negotiation is mandatory per IEEE for proper operation in 1000BASE-T. 1 = Enable Auto-Negotiation Process 0 = Disable Auto-Negotiation Process

Table 322: Copper Control Register (Continued)
Page 0, Register 0

Bits	Field	Mode	HW Rst	SW Rst	Description
11	Power Down	R/W	See Descr	Retain	<p>Power down is controlled via register 0_0.11 and 16_0.2. Both bits must be set to 0 before the PHY will transition from power down to normal operation.</p> <p>When the port is switched from power down to normal operation, software reset and restart Auto-Negotiation are performed even when bits Reset (0_0.15) and Restart Auto-Negotiation (0_0.9) are not set by the user.</p> <p>1 = Power down 0 = Normal operation</p>
10	Isolate	RO	0x0	0x0	This bit has no effect.
9	Restart Copper Auto-Negotiation	R/W, SC	0x0	SC	<p>Auto-Negotiation automatically restarts after hardware or software reset regardless of whether or not the restart bit (0.9) is set.</p> <p>1 = Restart Auto-Negotiation Process 0 = Normal operation</p>
8	Copper Duplex Mode	R/W	0x1	Update	<p>Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect.</p> <p>A write to this register bit does not take effect until any one of the following also occurs:</p> <ul style="list-style-type: none"> Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down <p>1 = Full-duplex 0 = Half-duplex</p>
7	Collision Test	RO	0x0	0x0	This bit has no effect.
6	Speed Selection (MSB)	R/W	0x1	Update	<p>Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect.</p> <p>A write to this register bit does not take effect until any one of the following occurs:</p> <ul style="list-style-type: none"> Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation bit 6, 13 Copper link goes down <p>11 = Reserved 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps</p>
5:0	Reserved	RO	Always 000000	Always 000000	Reserved



Link Street®

88E6321/88E6320 Functional Specification

Table 323: Copper Status Register
Page 0, Register 1

Bits	Field	Mode	HW Rst	SW Rst	Description
15	100BASE-T4	RO	Always 0	Always 0	100BASE-T4. This protocol is not available. 0 = PHY not able to perform 100BASE-T4
14	100BASE-X Full-Duplex	RO	Always 1	Always 1	1 = PHY able to perform full-duplex 100BASE-X
13	100BASE-X Half-Duplex	RO	Always 1	Always 1	1 = PHY able to perform half-duplex 100BASE-X
12	10 Mbps Full-Duplex	RO	Always 1	Always 1	1 = PHY able to perform full-duplex 10BASE-T
11	10 Mbps Half-Duplex	RO	Always 1	Always 1	1 = PHY able to perform half-duplex 10BASE-T
10	100BASE-T2 Full-Duplex	RO	Always 0	Always 0	This protocol is not available. 0 = PHY not able to perform full-duplex
9	100BASE-T2 Half-Duplex	RO	Always 0	Always 0	This protocol is not available. 0 = PHY not able to perform half-duplex
8	Extended Status	RO	Always 1	Always 1	1 = Extended status information in Register 15
7	Reserved	RO	Always 0	Always 0	Reserved
6	MF Preamble Suppression	RO	Always 1	Always 1	1 = PHY accepts management frames with preamble suppressed
5	Copper Auto-Negotiation Complete	RO	0x0	0x0	1 = Auto-Negotiation process complete 0 = Auto-Negotiation process not complete
4	Copper Remote Fault	RO,LH	0x0	0x0	1 = Remote fault condition detected 0 = Remote fault condition not detected
3	Auto-Negotiation Ability	RO	Always 1	Always 1	1 = PHY able to perform Auto-Negotiation
2	Copper Link Status	RO,LL	0x0	0x0	This register bit indicates when link was down since the last read. For the current link status, either read this register back-to-back or read Register 17_0.10 Link Real Time. 1 = Link is up 0 = Link is down
1	Jabber Detect	RO,LH	0x0	0x0	1 = Jabber condition detected 0 = Jabber condition not detected
0	Extended Capability	RO	Always 1	Always 1	1 = Extended register capabilities

Table 324: PHY Identifier 1
Page 0, Register 2

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Organizationally Unique Identifier Bit 3:18	RO	0x0141	0x0141	<p>Marvell® OUI is 0x005043</p> <p>0000 0000 0101 0000 0100 0011</p> <p> ^ ^</p> <p>bit 1.....bit 24</p> <p>Register 2.[15:0] show bits 3 to 18 of the OUI.</p> <p>0000000101000001</p> <p> ^ ^</p> <p>bit 3.....bit18</p>

Table 325: PHY Identifier 2
Page 0, Register 3

Bits	Field	Mode	HW Rst	SW Rst	Description
15:10	OUI Lsb	RO	Always 000011	Always 000011	Organizationally Unique Identifier bits 19:24 00 0011 ^.....^ bit 19...bit24
9:4	Model Number	RO	See Descr	See Descr	Contact Marvell® FAEs for information on the device Model Number.
3:0	Revision Number	RO	See Descr	See Descr	Contact Marvell® FAEs for information on the Device Revision Number.



Link Street®

88E6321/88E6320 Functional Specification

Table 326: Copper Auto-Negotiation Advertisement Register
Page 0, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	R/W	0x0	Update	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. If 1000BASE-T is advertised then the required next pages are automatically transmitted. Register 4.15 should be set to 0 if no additional next pages are needed. 1 = Advertise 0 = Not advertised
14	Ack	RO	Always 0	Always 0	Must be 0.
13	Remote Fault	R/W	0x0	Update	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. 1 = Set Remote Fault bit 0 = Do not set Remote Fault bit
12	Reserved	R/W	0x0	Update	Reserved
11	Asymmetric Pause	R/W	See Descr.	Update	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. 1 = Asymmetric Pause 0 = No asymmetric Pause
10	Pause	R/W	See Descr.	Update	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. 1 = Pause 0 = No pause
9	100BASE-T4	R/W	0x0	Retain	0 = Not capable of 100BASE-T4

Table 326: Copper Auto-Negotiation Advertisement Register (Continued)
Page 0, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
8	100BASE-TX Full-Duplex	R/W	See Descr.	Update	<p>A write to this register bit does not take effect until any one of the following occurs:</p> <p>Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down.</p> <p>If register 0_0.12 is set to 0 and speed is manually forced to 1000 Mbps in Registers 0_0.13 and 0_0.6, then Auto-Negotiation will still be enabled and only 1000BASE-T full-duplex is advertised if register 0_0.8 is set to 1, and 1000BASE-T half-duplex is advertised if 0_0.8 set to 0. Registers 4_0.8:5 and 9_0.9:8 are ignored. Auto-Negotiation is mandatory per IEEE for proper operation in 1000BASE-T.</p> <p>1 = Advertise 0 = Not advertised</p>
7	100BASE-TX Half-Duplex	R/W	See Descr.	Update	<p>A write to this register bit does not take effect until any one of the following occurs:</p> <p>Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down.</p> <p>If register 0_0.12 is set to 0 and speed is manually forced to 1000 Mbps in Registers 0_0.13 and 0_0.6, then Auto-Negotiation will still be enabled and only 1000BASE-T full-duplex is advertised if register 0_0.8 is set to 1, and 1000BASE-T half-duplex is advertised if 0_0.8 set to 0. Registers 4_0.8:5 and 9_0.9:8 are ignored. Auto-Negotiation is mandatory per IEEE for proper operation in 1000BASE-T.</p> <p>1 = Advertise 0 = Not advertised</p>
6	10BASE-TX Full-Duplex	R/W	See Descr.	Update	<p>A write to this register bit does not take effect until any one of the following occurs:</p> <p>Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down.</p> <p>If register 0_0.12 is set to 0 and speed is manually forced to 1000 Mbps in Registers 0_0.13 and 0_0.6, then Auto-Negotiation will still be enabled and only 1000BASE-T full-duplex is advertised if register 0_0.8 is set to 1, and 1000BASE-T half-duplex is advertised if 0_0.8 set to 0. Registers 4_0.8:5 and 9_0.9:8 are ignored. Auto-Negotiation is mandatory per IEEE for proper operation in 1000BASE-T.</p> <p>1 = Advertise 0 = Not advertised</p>



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 326: Copper Auto-Negotiation Advertisement Register (Continued)
Page 0, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
5	10BASE-TX Half-Duplex	R/W	See Descr.	Update	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. If register 0_0.12 is set to 0 and speed is manually forced to 1000 Mbps in Registers 0_0.13 and 0_0.6, then Auto-Negotiation will still be enabled and only 1000BASE-T full-duplex is advertised if register 0_0.8 is set to 1, and 1000BASE-T half-duplex is advertised if 0_0.8 set to 0. Registers 4_0.8:5 and 9_0.9:8 are ignored. Auto-Negotiation is mandatory per IEEE for proper operation in 1000BASE-T. 1 = Advertise 0 = Not advertised
4:0	Selector Field	R/W	0x01	Retain	Selector Field mode 00001 = 802.3

Table 327: Copper Link Partner Ability Register - Base Page
Page 0, Register 5

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	RO	0x0	0x0	Received Code Word Bit 15 1 = Link partner capable of next page 0 = Link partner not capable of next page
14	Acknowledge	RO	0x0	0x0	Acknowledge Received Code Word Bit 14 1 = Link partner received link code word 0 = Link partner does not have Next Page ability
13	Remote Fault	RO	0x0	0x0	Remote Fault Received Code Word Bit 13 1 = Link partner detected remote fault 0 = Link partner has not detected remote fault
12	Technology Ability Field	RO	0x0	0x0	Received Code Word Bit 12
11	Asymmetric Pause	RO	0x0	0x0	Received Code Word Bit 11 1 = Link partner requests asymmetric pause 0 = Link partner does not request asymmetric pause
10	Pause Capable	RO	0x0	0x0	Received Code Word Bit 10 1 = Link partner is capable of pause operation 0 = Link partner is not capable of pause operation
9	100BASE-T4 Capability	RO	0x0	0x0	Received Code Word Bit 9 1 = Link partner is 100BASE-T4 capable 0 = Link partner is not 100BASE-T4 capable
8	100BASE-TX Full-Duplex Capability	RO	0x0	0x0	Received Code Word Bit 8 1 = Link partner is 100BASE-TX full-duplex capable 0 = Link partner is not 100BASE-TX full-duplex capable
7	100BASE-TX Half-Duplex Capability	RO	0x0	0x0	Received Code Word Bit 7 1 = Link partner is 100BASE-TX half-duplex capable 0 = Link partner is not 100BASE-TX half-duplex capable
6	10BASE-T Full-Duplex Capability	RO	0x0	0x0	Received Code Word Bit 6 1 = Link partner is 10BASE-T full-duplex capable 0 = Link partner is not 10BASE-T full-duplex capable
5	10BASE-T Half-Duplex Capability	RO	0x0	0x0	Received Code Word Bit 5 1 = Link partner is 10BASE-T half-duplex capable 0 = Link partner is not 10BASE-T half-duplex capable
4:0	Selector Field	RO	0x00	0x00	Selector Field Received Code Word Bit 4:0



Link Street®

88E6321/88E6320 Functional Specification

**Table 328: Copper Auto-Negotiation Expansion Register
Page 0, Register 6**

Bits	Field	Mode	HW Rst	SW Rst	Description
15:5	Reserved	RO	0x000	0x000	Reserved.
4	Parallel Detection Fault	RO,LH	0x0	0x0	Register 6_0.4 is not valid until the Auto-Negotiation complete bit (Reg 1_0.5) indicates completed. 1 = A fault has been detected via the Parallel Detection function 0 = A fault has not been detected via the Parallel Detection function
3	Link Partner Next page Able	RO	0x0	0x0	Register 6_0.3 is not valid until the Auto-Negotiation complete bit (Reg 1_0.5) indicates completed. 1 = Link Partner is Next Page able 0 = Link Partner is not Next Page able
2	Local Next Page Able	RO	0x1	0x1	Register 6_0.2 is not valid until the Auto-Negotiation complete bit (Reg 1_0.5) indicates completed. 1 = Local Device is Next Page able 0 = Local Device is not Next Page able
1	Page Received	RO, LH	0x0	0x0	Register 6_0.1 is not valid until the Auto-Negotiation complete bit (Reg 1_0.5) indicates completed. 1 = A New Page has been received 0 = A New Page has not been received
0	Link Partner Auto-Negotiation Able	RO	0x0	0x0	Register 6_0.0 is not valid until the Auto-Negotiation complete bit (Reg 1_0.5) indicates completed. 1 = Link Partner is Auto-Negotiation able 0 = Link Partner is not Auto-Negotiation able

**Table 329: Copper Next Page Transmit Register
Page 0, Register 7**

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	R/W	0x0	0x0	A write to register 7_0 implicitly sets a variable in the Auto-Negotiation state machine indicating that the next page has been loaded. Link fail will clear Reg 7_0. Transmit Code Word Bit 15
14	Reserved	RO	0x0	0x0	Reserved
13	Message Page Mode	R/W	0x1	0x1	Transmit Code Word Bit 13
12	Acknowledge2	R/W	0x0	0x0	Transmit Code Word Bit 12
11	Toggle	RO	0x0	0x0	Transmit Code Word Bit 11
10:0	Message/ Unformatted Field	R/W	0x001	0x001	Transmit Code Word Bit 10:0

Table 330: Copper Link Partner Next Page Register
Page 0, Register 8

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	RO	0x0	0x0	Received Code Word Bit 15
14	Acknowledge	RO	0x0	0x0	Received Code Word Bit 14
13	Message Page	RO	0x0	0x0	Received Code Word Bit 13
12	Acknowledge2	RO	0x0	0x0	Received Code Word Bit 12
11	Toggle	RO	0x0	0x0	Received Code Word Bit 11
10:0	Message/ Unformatted Field	RO	0x000	0x000	Received Code Word Bit 10:0

Table 331: 1000BASE-T Control Register
Page 0, Register 9

Bits	Field	Mode	HW Rst	SW Rst	Description
15:13	Test Mode	R/W	0x0	Retain	<p>TX_CLK comes from the RX_CLK pin for jitter testing in test modes 2 and 3. After exiting the test mode, hardware reset or software reset (Register 0_0.15) should be issued to ensure normal operation. A restart of Auto-Negotiation will clear these bits.</p> <p>000 = Normal Mode 001 = Test Mode 1 - Transmit Waveform Test 010 = Test Mode 2 - Transmit Jitter Test (MASTER mode) 011 = Test Mode 3 - Transmit Jitter Test (SLAVE mode) 100 = Test Mode 4 - Transmit Distortion Test 101, 110, 111 = Reserved</p>
12	MASTER/SLAVE Manual Configuration Enable	R/W	0x0	Update	<p>A write to this register bit does not take effect until any of the following also occurs:</p> <p>Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down.</p> <p>1 = Manual MASTER/SLAVE configuration 0 = Automatic MASTER/SLAVE configuration</p>
11	MASTER/SLAVE Configuration Value	R/W	See Descr.	Update	<p>A write to this register bit does not take effect until any of the following also occurs:</p> <p>Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down.</p> <p>1 = Manual configure as MASTER 0 = Manual configure as SLAVE</p>



Table 331: 1000BASE-T Control Register (Continued)

Page 0, Register 9

Bits	Field	Mode	HW Rst	SW Rst	Description
10	Port Type	R/W	See Descr.	Update	A write to this register bit does not take effect until any of the following also occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. Register 9_0.10 is ignored if Register 9_0.12 is equal to 1. 1 = Prefer multi-port device (MASTER) 0 = Prefer single port device (SLAVE)
9	1000BASE-T Full-Duplex	R/W	0x1	Update	A write to this register bit does not take effect until any of the following also occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Link goes down 1 = Advertise 0 = Not advertised
8	1000BASE-T Half-Duplex	R/W	See Descr.	Update	A write to this register bit does not take effect until any of the following also occurs: Software reset is asserted (Register 0_0.15) Restart Auto-Negotiation is asserted (Register 0_0.9) Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation Copper link goes down. 1 = Advertise 0 = Not advertised
7:0	Reserved	R/W	0x00	Retain	Reserved

Table 332: 1000BASE-T Status Register

Page 0, Register 10

Bits	Field	Mode	HW Rst	SW Rst	Description
15	MASTER/SLAVE Configuration Fault	RO,LH	0x0	0x0	This register bit will clear on read. 1 = MASTER/SLAVE configuration fault detected 0 = No MASTER/SLAVE configuration fault detected
14	MASTER/SLAVE Configuration Resolution	RO	0x0	0x0	1 = Local PHY configuration resolved to MASTER 0 = Local PHY configuration resolved to SLAVE
13	Local Receiver Status	RO	0x0	0x0	1 = Local Receiver OK 0 = Local Receiver is Not OK
12	Remote Receiver Status	RO	0x0	0x0	1 = Remote Receiver OK 0 = Remote Receiver Not OK
11	Link Partner 1000BASE-T Full-Duplex Capability	RO	0x0	0x0	1 = Link Partner is capable of 1000BASE-T full-duplex 0 = Link Partner is not capable of 1000BASE-T full-duplex

Table 332: 1000BASE-T Status Register (Continued)

Page 0, Register 10

Bits	Field	Mode	HW Rst	SW Rst	Description
10	Link Partner 1000BASE-T Half-Duplex Capability	RO	0x0	0x0	1 = Link Partner is capable of 1000BASE-T half-duplex 0 = Link Partner is not capable of 1000BASE-T half-duplex
9:8	Reserved	RO	0x0	0x0	Reserved
7:0	Idle Error Count	RO, SC	0x00	0x00	MSB of Idle Error Counter These register bits report the idle error count since the last time this register was read. The counter pegs at 11111111 and will not roll over.

Table 333: XMDIO MMD Access Control Register

Page 0, Register 13

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Function	R/W	0x0	Retain	15:14 00 = address 01 = data, no post increment 10 = data, post increment on reads and writes 11 = data, post increment on writes only
13:5	Reserved	RO	0x000	Retain	Reserved
4:0	DEVAD	RO	0x00	Retain	Device address

Table 334: XMDIO MMD Access Address/Data Register

Page 0, Register 14

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Address Data	R/W	0x0000	See Descr.	If 13.15:14 = 00, this shows the address of the register with the MMD device address as indicated by 13_4:0. For this setting, if a software reset occurs, the address is retained.



Table 335: Extended Status Register
Page 0, Register 15

Bits	Field	Mode	HW Rst	SW Rst	Description
15	1000BASE-X Full-Duplex	RO	Always 0	Always 0	0 = not 1000BASE-X full-duplex capable
14	1000BASE-X Half-Duplex	RO	Always 0	Always 0	0 = not 1000BASE-X half-duplex capable
13	1000BASE-T Full-Duplex	RO	Always 1	Always 1	1 = 1000BASE-T full-duplex capable
12	1000BASE-T Half-Duplex	RO	Always 1	Always 1	1 = 1000BASE-T half-duplex capable
11:0	Reserved	RO	0x000	0x000	Reserved

Table 336: Copper Specific Control Register 1
Page 0, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Disable Link Pulses	R/W	0x0	0x0	1 = Disable Link Pulse 0 = Enable Link Pulse
14:12	Downshift counter	R/W	0x3	Update	Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by software reset to take effect. 1x, 2x, ...8x is the number of times the PHY attempts to establish Gigabit link before the PHY downshifts to the next highest speed. 000 = 1x 100 = 5x 001 = 2x 101 = 6x 010 = 3x 110 = 7x 011 = 4x 111 = 8x
11	Downshift Enable	R/W	0x0	Update	Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by software reset to take effect. 1 = Enable downshift. 0 = Disable downshift.
10	Force Copper Link Good	R/W	0x0	Retain	If link is forced to be good, the link state machine is bypassed and the link is always up. In 1000BASE-T mode this has no effect. 1 = Force link good 0 = Normal operation
9:7	Energy Detect	R/W	See Descr.	Update	0xx = Off 100 = sense only on receiver (Energy Detect), auto wake-up, 101 = sense only on receiver (Energy Detect), SW wake-up 110 = sense and periodically transmit NLP (Energy Detect+TM), auto wake-up 111 = sense and periodically transmit NLP (Energy Detect+TM), SW wake-up

Table 336: Copper Specific Control Register 1 (Continued)

Page 0, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
6:5	MDI Crossover Mode	R/W	See Descr.	Update	<p>Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect.</p> <p>00 = Manual MDI configuration 01 = Manual MDIX configuration 10 = Reserved 11 = Enable automatic crossover for all modes</p>
4	Energy Detect wake up control	R/W or RO, SC	0x0	0	<p>This bit controls how the PHY wake up from Energy detect state.</p> <p>If 16_0.7 = 1 (SW wake-up), this register bit is R/W. SW write of 1 to this bit will wake up the PHY. Then the bit will self clear after the PHY wakes up.</p> <p>1 = Sleep 0 = Active</p> <p>If 16_0.7 = 0, this register bit is RO and reflects wake up status. The PHY will wake up from energy detect state automatically based on the energy detected from line.</p>
3	Copper Transmitter Disable	R/W	0x0	Retain	1 = Transmitter Disable 0 = Transmitter Enable
2	Power Down	R/W	0x0	Retain	<p>Power down is controlled via register 0_0.11 and 16_0.2. Both bits must be set to 0 before the PHY will transition from power down to normal operation.</p> <p>When the port is switched from power down to normal operation, software reset and restart Auto-Negotiation are performed even when bits Reset (0_0.15) and Restart Auto-Negotiation (0_0.9) are not set by the user.</p> <p>Upon hardware reset this bit takes on the value of phyg_pwrdrn_a[0].</p> <p>1 = Power down 0 = Normal operation</p>
1	Polarity Reversal Disable	R/W	0x0	Retain	<p>If polarity is disabled, then the polarity is forced to be normal in 10BASE-T.</p> <p>1 = Polarity Reversal Disabled 0 = Polarity Reversal Enabled</p> <p>The detected polarity status is shown in Register 17_0.1, or in 1000BASE-T mode, 21_5.3:0.</p>
0	Disable Jabber	R/W	0x0	Retain	Jabber has effect only in 10BASE-T half-duplex mode. 1 = Disable jabber function 0 = Enable jabber function



Link Street®

88E6321/88E6320 Functional Specification

Table 337: Copper Specific Status Register 1
Page 0, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Speed	RO	0x2	Retain	<p>These status bits are valid only after resolved bit 17_0.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled.</p> <p>11 = Reserved 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps</p>
13	Duplex	RO	0x0	Retain	<p>This status bit is valid only after resolved bit 17_0.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled.</p> <p>1 = Full-duplex 0 = Half-duplex</p>
12	Page Received	RO, LH	0x0	0x0	1 = Page received 0 = Page not received
11	Speed and Duplex Resolved	RO	0x0	0x0	When Auto-Negotiation is not enabled 17_0.11 = 1. 1 = Resolved 0 = Not resolved
10	Copper Link (real time)	RO	0x0	0x0	1 = Link up 0 = Link down
9	Transmit Pause Enabled	RO	0x0	0x0	<p>This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the device.</p> <p>This status bit is valid only after resolved bit 17_0.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled.</p> <p>1 = Transmit pause enabled 0 = Transmit pause disable</p>
8	Receive Pause Enabled	RO	0x0	0x0	<p>This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the device.</p> <p>This status bit is valid only after resolved bit 17_0.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled.</p> <p>1 = Receive pause enabled 0 = Receive pause disabled</p>
7	Reserved	RO	0x0	0x0	Reserved
6	MDI Crossover Status	RO	0x1	Retain	<p>This status bit is valid only after resolved bit 17_0.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. This bit is 0 or 1 depending on what is written to 16_0.6:5 in manual configuration mode. Register 16_0.6:5 are updated with software reset.</p> <p>1 = MDIX 0 = MDI</p>
5	Downshift Status	RO	0x0	0x0	1 = Downshift 0 = No Downshift
4	Copper Energy Detect Status	RO	0x0	0x0	1 = Sleep 0 = Active NOTE: If 16_0.7 = 1, (SW wake-up mode - 16_0.4 = 0) and energy is received, this bit will be 1.

Table 337: Copper Specific Status Register 1 (Continued)
Page 0, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
3	Global Link Status	RO	0x0	0x0	1 = Copper link is up 0 = Copper link is down
2	DTE power status	RO	0x0	0x0	1 = Link partner needs DTE power 0 = Link partner does not need DTE power
1	Polarity (real time)	RO	0x0	0x0	1 = Reversed 0 = Normal Polarity reversal can be disabled by writing to Register 16_0.1. In 1000BASE-T mode, polarity of all pairs are shown in Register 21_5.3:0.
0	Jabber (real time)	RO	0x0	0x0	1 = Jabber 0 = No jabber



Link Street®

88E6321/88E6320 Functional Specification

Table 338: Copper Specific Interrupt Enable Register
Page 0, Register 18

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Auto-Negotiation Error Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
14	Speed Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
13	Duplex Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
12	Page Received Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
11	Auto-Negotiation Completed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
10	Link Status Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
9	Symbol Error Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
8	False Carrier Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
7	Reserved	R/W	0x0	Retain	Reserved
6	MDI Crossover Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
5	Downshift Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
4	Copper Energy Detect Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
3	FLP Exchange Complete but no Link Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
2	Reserved	R/W	0x0	Retain	Reserved for future use. This bit must be 0.
1	Polarity Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
0	Jabber Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable

Table 339: Copper Interrupt Status Register
Page 0, Register 19

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Copper Auto-Negotiation Error	RO,LH	0x0	0x0	An error is said to occur if MASTER/SLAVE does not resolve, parallel detect fault, no common HCD, or link does not come up after negotiation is completed. 1 = Auto-Negotiation Error 0 = No Auto-Negotiation Error
14	Copper Speed Changed	RO,LH	0x0	0x0	1 = Speed changed 0 = Speed not changed
13	Copper Duplex Changed	RO,LH	0x0	0x0	1 = Duplex changed 0 = Duplex not changed
12	Copper Page Received	RO,LH	0x0	0x0	1 = Page received 0 = Page not received
11	Copper Auto-Negotiation Completed	RO,LH	0x0	0x0	1 = Auto-Negotiation completed 0 = Auto-Negotiation not completed
10	Copper Link Status Changed	RO,LH	0x0	0x0	1 = Link status changed 0 = Link status not changed
9	Copper Symbol Error	RO,LH	0x0	0x0	1 = Symbol error 0 = No symbol error
8	Copper False Carrier	RO,LH	0x0	0x0	1 = False carrier 0 = No false carrier
7	Reserved	RO	Always 0	Always 0	Reserved
6	MDI Crossover Changed	RO,LH	0x0	0x0	1 = Crossover changed 0 = Crossover not changed
5	Downshift Interrupt	RO,LH	0x0	0x0	1 = Downshift detected 0 = No down shift
4	Copper Energy Detect Changed	RO,LH	0x0	0x0	1 = Energy Detect state changed 0 = No Energy Detect state change detected
3	FLP Exchange Complete but no Link	RO,LH	0x0	0x0	1 = FLP Exchange Completed but Link Not Established 0 = No Event Detected
2	DTE power detection status changed interrupt	RO,LH	0x0	0x0	1 = DTE power detection status changed 0 = No DTE power detection status change detected
1	Polarity Changed	RO,LH	0x0	0x0	1 = Polarity Changed 0 = Polarity not changed
0	Jabber	RO,LH	0x0	0x0	1 = Jabber 0 = No jabber



Link Street®

88E6321/88E6320 Functional Specification

Table 340: Copper Specific Control Register 2
Page 0, Register 20

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Low Power Transmitter Enable	R/W	0x1	Retain	0 = Transmitter is always on 1 = Transmitter turns off in between FLP burst transmissions during Auto-Negotiation to save power.
14:8	Reserved	R/W	0x000	Retain	Reserved
7	10BASE-Te Enable	R/W	0	Retain	Upon hardware reset this bit takes on the value of pg_config_10bte_a. 0 = Disable 10BASE-Te 1 = Enable 10BASE-Te
6	Break Link On Insufficient IPG	R/W	0x0	Retain	0 = Break link on insufficient IPGs in 10BASE-T and 100BASE-TX. 1 = Do not break link on insufficient IPGs in 10BASE-T and 100BASE-TX.
5	100BASE-T Transmitter Clock Source	R/W	0x1	Update	1 = Local Clock 0 = Recovered Clock
4	Accelerate 100BASE-T Link Up	R/W	0x0	Retain	0 = No Acceleration 1 = Accelerate
3	Reverse MDIP/N[3] Transmit Polarity	R/W	0x0	Retain	0 = Normal Transmit Polarity, 1 = Reverse Transmit Polarity
2	Reverse MDIP/N[2] Transmit Polarity	R/W	0x0	Retain	0 = Normal Transmit Polarity, 1 = Reverse Transmit Polarity
1	Reverse MDIP/N[1] Transmit Polarity	R/W	0x0	Retain	0 = Normal Transmit Polarity, 1 = Reverse Transmit Polarity
0	Reverse MDIP/N[0] Transmit Polarity	R/W	0x0	Retain	0 = Normal Transmit Polarity, 1 = Reverse Transmit Polarity

Table 341: Copper Specific Receive Error Counter Register
Page 0, Register 21

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Receive Error Count	ROC	0x0000	Retain	Counter will peg at 0xFFFF and will not roll over. Both False carrier and symbol errors are reported.

Table 342: Page Address
Page Any, Register 22

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Reserved	R/W	0x0	Retain	Reserved. These bits must be 0.
13:8	Reserved	RO	0x00	0x00	Reserved
7:0	Page select for registers 0 to 28	R/W	0x00	Retain	Page Number

Table 343: Global Interrupt Status
Page 0, Register 23

Bits	Field	Mode	HW Rst	SW Rst	Description
15:5	Reserved	RO	Always 0s	Always 0s	0
4	Port 4 Interrupt	RO	0x0	0x0	If there are at least 5 copper ports, then this bit indicates Port 4's interrupt status; otherwise, it is invalid. 1 = Interrupt active on Port 4 0 = No interrupt active on Port 4
3	Port 3 Interrupt	RO	0x0	0x0	If there are at least 4 copper ports, then this bit indicates Port 3's interrupt status; otherwise, it is invalid. 1 = Interrupt active on Port 3 0 = No interrupt active on Port 3
2	Port 2 Interrupt	RO	0x0	0x0	If there are at least 3 copper ports, then this bit indicates Port 2's interrupt status; otherwise, it is invalid. 1 = Interrupt active on Port 2 0 = No interrupt active on Port 2
1	Port 1 Interrupt	RO	0x0	0x0	If there are at least 2 copper ports, then this bit indicates Port 1's interrupt status; otherwise, it is invalid. 1 = Interrupt active on Port 1 0 = No interrupt active on Port 1
0	Port 0 Interrupt	RO	0x0	0x0	If there is at least 1 copper port, then this bit indicates Port 0's interrupt status; otherwise, it is invalid. 1 = Interrupt active on Port 0 0 = No interrupt active on Port 0



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 344: Copper Specific Control Register 3
Page 0, Register 26

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	R/W	0	Retain	Reserved
14	Disable 1000BASE-T	R/W	See Desc.	Retain	<p>When set to disabled, 1000BASE-T will not be advertised even if registers 9_0.9 or 9_0.8 are set to 1.</p> <p>A write to this register bit does not take effect until any one of the following occurs:</p> <ul style="list-style-type: none">• Software reset is asserted (Register 0_0.15)• Restart Auto-Negotiation is asserted (Register 0_0.9)• Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation• Copper link goes down. <p>1 = Disable 1000BASE-T Advertisement 0 = Enable 1000BASE-T Advertisement</p>
13	Reverse Autoneg	R/W	See Desc.	Retain	<p>A write to this register bit does not take effect until any one of the following occurs:</p> <ul style="list-style-type: none">• Software reset is asserted (Register 0_0.15)• Restart Auto-Negotiation is asserted (Register 0_0.9)• Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation• Copper link goes down. <p>1 = Reverse Auto-Negotiation 0 = Normal Auto-Negotiation</p>
12	Disable 100BASE-T	R/W	See Descr.	Retain	<p>When set to disabled, 100BASE-TX will not be advertised even if registers 4_0.8 or 4_0.7 are set 1.</p> <p>A write to this register bit does not take effect until any one of the following occurs:</p> <ul style="list-style-type: none">• Software reset is asserted (Register 0_0.15)• Restart Auto-Negotiation is asserted (Register 0_0.9)• Power down (Register 0_0.11, 16_0.2) transitions from power down to normal operation• Copper link goes down. <p>1 = Disable 100BASE-TX Advertisement 0 = Enable 100BASE-TX Advertisement</p>
11:10	Gigabit Link Down Delay	R/W	0x0	Retain	This register only have effect if register 26_0.9 is set to 1. 00 = 0ms 01 = 10 ± 2ms 10 = 20 ± 2ms 11 = 40 ± 2ms
9	Speed Up Gigabit Link Down Time	R/W	0x0	Retain	1 = Enable faster gigabit link down 0 = Use IEEE gigabit link down
8	DTE detect enable	R/W	0x0	Update	1 = Enable DTE detection 0 = Disable DTE detection
7:4	DTE detect status drop hysteresis	R/W	0x4	Retain	0000: report immediately 0001: report 5s after DTE power status drop ... 1111: report 75s after DTE power status drop

Table 344: Copper Specific Control Register 3

Page 0, Register 26

Bits	Field	Mode	HW Rst	SW Rst	Description
3:2	100 MB test select	R/W	0x0	Retain	0x = Normal Operation 10 = Select 112 ns sequence 11 = Select 16 ns sequence
1	10 BT polarity force	R/W	0x0	Retain	1 = Force negative polarity for Receive only 0 = Normal Operation
0	Reserved	R/W	0x0	Retain	Reserved



Link Street®

88E6321/88E6320 Functional Specification

Table 345: MAC Specific Control Register 1
Page 2, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Copper Transmit FIFO Depth	R/W	See Descr.	Retain	00 = Handles 1518 byte packets 01 = Handles 9K byte packets 10 = Handles 18K byte packets 11 = Handles 27K byte packets
13:11	Reserved	R/W	0x4	Retain	Reserved
10	RCLK enable	R/W	0x0	Retain	1 = Enable RCLK 0 = Disable RCLK This bit must be set to one when recovering clock from this port.
9:8	Reserved	R/W	0x0	Retain	Reserved
7	Copper Reference Clock Source Select	R/W	See Descr	Update	Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect. 1 = Use SE_SCLK as 25 MHz source 0 = Use XTAL_IN as 25 MHz source NOTE: This bit only applies to multi-port IP
6	Pass Odd Nibble Preambles	R/W	0x1	Update	0 = Pad odd nibble preambles in copper receive packets. 1 = Pass as is and do not pad odd nibble preambles in copper receive packets.
5	DPLL Reference Clock Source Select	R/W	0x0	Retain	1 = Use SE_SCLK as the DPLL's 25 MHz source' 0 = Use XTAL_IN as the DPLL's 25 MHz source
4	Reserved	R/W	0x0	Retain	Write 0
3	MAC Interface Power Down	R/W	0x1	Update	Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect. This bit determines whether the MAC Interface powers down when Register 0_0.11, 16_0.2 are used to power down the device or when the PHY enters the energy detect state. 1 = Always power up 0 = Can power down
2:0	Reserved	R/W	0x0	Retain	Reserved

Table 346: MAC Specific Interrupt Enable Register
Page 2, Register 18

Bits	Field	Mode	HW Rst	SW Rst	Description
15:8	Reserved	R/W	0x00	Retain	00000000
7	FIFO Over/Underflow Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
6:4	Reserved	R/W	0x0	Retain	000
3	FIFO Idle Inserted Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
2	FIFO Idle Deleted Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
1:0	Reserved	R/W	0x0	Retain	00



Table 347: MAC Specific Status Register
Page 2, Register 19

Bits	Field	Mode	HW Rst	SW Rst	Description
15:8	Reserved	RO	Always 00	Always 00	Reserved
7	FIFO Over/Underflow	RO,LH	0x0	0x0	1 = Over/Underflow Error 0 = No FIFO Error
6:4	Reserved	RO	Always 0	Always 0	Reserved
3	FIFO Idle Inserted	RO,LH	0x0	0x0	1 = Idle Inserted 0 = No Idle Inserted
2	FIFO Idle Deleted	RO,LH	0x0	0x0	1 = Idle Deleted 0 = Idle not Deleted
1:0	Reserved	RO	Always 0	Always 0	Reserved

Table 348: Copper RX_ER Byte Capture
Page 2, Register 20

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Capture Data Valid	RO	0x0	0x0	1 = Bits 14:0 Valid 0 = Bits 14:0 Invalid
14	Reserved	RO	0x0	0x0	Reserved
13:12	Byte Number	RO	0x0	0x0	00 = 4 bytes before RX_ER asserted 01 = 3 bytes before RX_ER asserted 10 = 2 bytes before RX_ER asserted 11 = 1 byte before RX_ER asserted The byte number increments after every read when register 20_2.15 is set to 1.
11:10	Reserved	RO	0x0	0x0	Reserved
9	RX_ER	RO	0x0	0x0	RX Error. Normally this bit will be low since the capture is triggered by RX_ER being high. However it is possible to see an RX_ER high when the capture is re-enabled after reading the fourth byte and there happens to be a long sequence of RX_ER when the capture restarts.
8	RX_DV	RO	0x0	0x0	RX Data Valid
7:0	RXD[7:0]	RO	0x00	0x00	RX Data

Table 349: MAC Specific Control Register 2
Page 2, Register 21

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	R/W	0x0	0x0	Reserved
14	Copper Line Loopback	R/W	0x0	0x0	1 = Enable Loopback of MDI to MDI 0 = Normal Operation
13:12	Reserved	R/W	0x1	Update	Reserved
11:7	Reserved	R/W	0x00	0x00	Reserved
6	Reserved	R/W	0x1	Update	Reserved
5:4	Reserved	R/W	0x0	Update	Reserved
3	Block Carrier Extension Bit	R/W	0x0	Retain	1 = Enable Block Carrier Extension 0 = Disable Block Carrier Extension
2:0	Default MAC interface speed	R/W	0x6	Update	Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by software reset to take effect. MAC Interface Speed during Link down while Auto-Negotiation is enabled. Bit Speed 0XX = Reserved 100 = 10 Mbps 101 = 100 Mbps 110 = 1000 Mbps 111 = Reserved

Table 350: Advanced VCT™ TX to MDI[0] Rx Coupling
Page 5, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reflected Polarity	RO	xx	Retain	1 = Positive Reflection, 0 = negative reflection
14:8	Reflected Amplitude	RO	xx	Retain	<p>0000000 = No reflection (0 mV) each bit above increases 7.8125mV.</p> <p>When 23_5.7 = 0 and 23_5.13:11 = 000 or 100 the reflected amplitude between the thresholds specified in registers 26_5, 27_5, 28_5.6:0, 26_7, 27_7, and 28_7.6:0 are reported as 0 mV.</p> <p>When 23_5.7 = 0 and 23_5.13:11 is not 000 or 100 the reflected amplitude between the thresholds specified in register 25_5 and 25_7 are reported as 0 mV.</p> <p>When 23_5.7 = 1 the actual offset or reflected amplitude is reported and the threshold specified in registers 25_5, 26_5, 27_5, 28_5, 25_7, 26_7, 27_7, and 28_7 are ignored.</p> <p>The amplitude value is valid only When 23_5.14 = 1. If bit 15:8 = 0x00 indicates that the test failed.</p>
7:0	Distance	RO	xx	Retain	<p>Distance of reflection.</p> <p>The distance value is valid only when 23_5.7 = 0 and 23_5.14 = 1.</p>


Note

This register reports the reflection see based on the setting of register 23_5.13:11
000 = MDI[0] Tx to MDI[0] Rx
100 = MDI[0] Tx to MDI[0] Rx
101 = MDI[1] Tx to MDI[0] Rx
110 = MDI[2] Tx to MDI[0] Rx
111 = MDI[3] Tx to MDI[0] Rx

Table 351: Advanced VCT™ TX to MDI[1] Rx Coupling
Page 5, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reflected Polarity	RO	xx	Retain	1 = Positive Reflection, 0 = negative reflection
14:8	Reflected Amplitude	RO	xx	Retain	<p>0000000 = No reflection (0 mV) each bit above increases 7.8125mV.</p> <p>When 23_5.7 = 0 and 23_5.13:11 = 000 or 101 the reflected amplitude between the thresholds specified in registers 26_5, 27_5, 28_5.6:0, 26_7, 27_7, and 28_7.6:0 are reported as 0 mV.</p> <p>When 23_5.7 = 0 and 23_5.13:11 is not 000 or 101 the reflected amplitude between the thresholds specified in register 25_5 and 25_7 are reported as 0 mV.</p> <p>When 23_5.7 = 1 the actual offset or reflected amplitude is reported and the threshold specified in registers 25_5, 26_5, 27_5, 28_5, 25_7, 26_7, 27_7, and 28_7 are ignored.</p> <p>The amplitude value is valid only When 23_5.14 = 1. If bit 15:8 = 0x00 indicates that the test failed.</p>
7:0	Distance	RO	xx	Retain	<p>Distance of reflection.</p> <p>The distance value is valid only when 23_5.7 = 0 and 23_5.14 = 1.</p>

**Note**

This register reports the reflection see based on the setting of register 23_5.13:11
 000 = MDI[1] Tx to MDI[1] Rx
 100 = MDI[0] Tx to MDI[1] Rx
 101 = MDI[1] Tx to MDI[1] Rx
 110 = MDI[2] Tx to MDI[1] Rx
 111 = MDI[3] Tx to MDI[1] Rx



Link Street®

88E6321/88E6320 Functional Specification

Table 352: Advanced VCT™ TX to MDI[2] Rx Coupling
Page 5, Register 18

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reflected Polarity	RO	xx	Retain	1 = Positive Reflection, 0 = negative reflection
14:8	Reflected Amplitude	RO	xx	Retain	0000000 = No reflection (0 mV) each bit above increases 7.8125mV. When 23_5.7 = 0 and 23_5.13:11 = 000 or 110 the reflected amplitude between the thresholds specified in registers 26_5, 27_5, 28_5.6:0, 26_7, 27_7, and 28_7.6:0 are reported as 0 mV. When 23_5.7 = 0 and 23_5.13:11 is not 000 or 110 the reflected amplitude between the thresholds specified in register 25_5 and 25_7 are reported as 0 mV. When 23_5.7 = 1 the actual offset or reflected amplitude is reported and the threshold specified in registers 25_5, 26_5, 27_5, 28_5, 25_7, 26_7, 27_7, and 28_7 are ignored. The amplitude value is valid only When 23_5.14 = 1. If bit 15:8 = 0x00 indicates that the test failed.
7:0	Distance	RO	xx	Retain	Distance of reflection. The distance value is valid only when 23_5.7 = 0 and 23_5.14 = 1.

**Note**

This register reports the reflection see based on the setting of register 23_5.13:11
000 = MDI[2] Tx to MDI[2] Rx
100 = MDI[0] Tx to MDI[2] Rx
101 = MDI[1] Tx to MDI[2] Rx
110 = MDI[2] Tx to MDI[2] Rx
111 = MDI[3] Tx to MDI[2] Rx

Table 353: Advanced VCT™ TX to MDI[3] Rx Coupling
Page 5, Register 19

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reflected Polarity	RO	xx	Retain	1 = Positive Reflection, 0 = negative reflection
14:8	Reflected Amplitude	RO	xx	Retain	<p>0000000 = No reflection (0 mV) each bit above increases 7.8125mV.</p> <p>When 23_5.7 = 0 and 23_5.13:11 = 000 or 111 the reflected amplitude between the thresholds specified in registers 26_5, 27_5, 28_5.6:0, 26_7, 27_7, and 28_7.6:0 are reported as 0 mV.</p> <p>When 23_5.7 = 0 and 23_5.13:11 is not 000 or 111 the reflected amplitude between the thresholds specified in register 25_5 and 25_7 are reported as 0 mV.</p> <p>When 23_5.7 = 1 the actual offset or reflected amplitude is reported and the threshold specified in registers 25_5, 26_5, 27_5, 28_5, 25_7, 26_7, 27_7, and 28_7 are ignored.</p> <p>The amplitude value is valid only When 23_5.14 = 1. If bit 15:8 = 0x00 indicates that the test failed.</p>
7:0	Distance	RO	xx	Retain	<p>Distance of reflection.</p> <p>The distance value is valid only when 23_5.7 = 0 and 23_5.14 = 1.</p>

**Note**

This register reports the reflection see based on the setting of register 23_5.13:11
 000 = MDI[3] Tx to MDI[3] Rx
 100 = MDI[0] Tx to MDI[3] Rx
 101 = MDI[1] Tx to MDI[3] Rx
 110 = MDI[2] Tx to MDI[3] Rx
 111 = MDI[3] Tx to MDI[3] Rx



Link Street®

88E6321/88E6320 Functional Specification

Table 354: 1000BASE-T Pair Skew Register
Page 5, Register 20

Bits	Field	Mode	HW Rst	SW Rst	Description
15:12	Pair 7,8 (MDI[3]±)	RO	0x0	0x0	Skew = bit value x 8ns. Value is correct to within ± 8ns. The contents of 20_15:0 are valid only if Register 21_5.6 = 1
11:8	Pair 4,5 (MDI[2]±)	RO	0x0	0x0	Skew = bit value x 8ns. Value is correct to within ± 8ns.
7:4	Pair 3,6 (MDI[1]±)	RO	0x0	0x0	Skew = bit value x 8ns. Value is correct to within ± 8ns.
3:0	Pair 1,2 (MDI[0]±)	RO	0x0	0x0	Skew = bit value x 8ns. Value is correct to within ± 8ns.

Table 355: 1000BASE-T Pair Swap and Polarity
Page 5, Register 21

Bits	Field	Mode	HW Rst	SW Rst	Description
15:7	Reserved	RO	0x000	0x000	Reserved
6	Register 20_5 and 21_5 valid	RO	0x0	0x0	The contents of 21_5.5:0 and 20_5.15:0 are valid only if Register 21_5.6 = 1 1= Valid 0 = Invalid
5	C, D Crossover	RO	0x0	0x0	1 = Channel C received on MDI[2]± Channel D received on MDI[3]± 0 = Channel D received on MDI[2]± Channel C received on MDI[3]±
4	A, B Crossover	RO	0x0	0x0	1 = Channel A received on MDI[0]± Channel B received on MDI[1]± 0 = Channel B received on MDI[0]± Channel A received on MDI[1]±
3	Pair 7,8 (MDI[3]±) Polarity	RO	0x0	0x0	1 = Negative 0 = Positive
2	Pair 4,5 (MDI[2]±) Polarity		0x0	0x0	1 = Negative 0 = Positive
1	Pair 3,6 (MDI[1]±) Polarity	RO	0x0	0x0	1 = Negative 0 = Positive
0	Pair 1,2 (MDI[0]±) Polarity	RO	0x0	0x0	1 = Negative 0 = Positive

Table 356: Advanced VCT™ Control
Page 5, Register 23

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Enable Test	R/W, SC	0x0	0x0	0 = disable test, 1 = enable test This bit will self clear when the test is completed
14	Test status	RO	0x0	0x0	0 = Test not started/in progress, 1 = test completed
13:11	Transmitter Channel Select	R/W	0x0	0x0	000 - Tx 0 => Rx 0, Tx 1 => Rx 1, Tx 2 => Rx 2, Tx 3 => Rx 3. 100 - Tx 0 => Rx 0, Tx 0 => Rx 1, Tx 0 => Rx 2, Tx 0 => Rx 3. 101 - Tx 1 => Rx 0, Tx 1 => Rx 1, Tx 1 => Rx 2, Tx 1 => Rx 3. 110 - Tx 2 => Rx 0, Tx 2 => Rx 1, Tx 2 => Rx 2, Tx 2 => Rx 3. 111 - Tx 3 => Rx 0, Tx 3 => Rx 1, Tx 3 => Rx 2, Tx 3 => Rx 3. 01x - Reserved 0x1 - Reserved
10:8	Number of Sample Averaged	R/W	6	Retain	0 = 2 samples 1 = 4 samples 2 = 8 samples 3 = 16 samples 4 = 32 samples 5 = 64 samples 6 = 128 samples 7 = 256 samples
7:6	Mode	R/W	0x0	Retain	00 = Maximum peak above threshold 01 = First or last peak above threshold. See register 28_5.13. 10 = Offset 11 = Sample point at distance set by 24_5.7:0
5:0	Peak Detection Hysteresis	R/W	0x03	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x3F = ± 492 mV

Table 357: Advanced VCT™ Sample Point Distance
Page 5, Register 24

Bits	Field	Mode	HW Rst	SW Rst	Description
15:10	Reserved	RO	0x00	0x00	Reserved
9:0	Distance to measure/ Distance to start	R/W	0x000	Retain	When 23_5.7:6 = 11 the measurement is taken at this distance. (00 to 3FF) When 23_5.7:6 = 0x any distance below this distance is not considered (00 to FF). Bit 9:8 is ignored.



Link Street®

88E6321/88E6320 Functional Specification

Table 358: Advanced VCT™ Cross Pair Positive Threshold
Page 5, Register 25

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14:8	Cross Pair Positive Threshold > 30m	R/W	0x01	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = - 992 mV
7	Reserved	RO	0x0	0x0	Reserved
6:0	Cross Pair Positive Threshold < 30m	R/W	0x04	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = - 992 mV

Table 359: Advanced VCT Same Pair Impedance Positive Threshold 0 and 1
Page 5, Register 26

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14:8	Same-Pair Positive Threshold 10m - 50m	R/W	0x0F	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = - 992 mV
7	Reserved	RO	0x0	0x0	Reserved
6:0	Same-Pair Positive Threshold < 10m	R/W	0x12	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = - 992 mV

Table 360: Advanced VCT™ Same Pair Impedance Positive Threshold 2 and 3
Page 5, Register 27

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14:8	Same-Pair Positive Threshold 110m - 140m	R/W	0x0A	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = - 992 mV
7	Reserved	RO	0x0	0x0	Reserved
6:0	Same-Pair Positive Threshold 50m - 110m	R/W	0x0C	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = - 992 mV

Table 361: Advanced VCT™ Same Pair Impedance Positive Threshold 4 and Transmit Pulse Control
Page 5, Register 28

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Reserved	RO	0x0	0x0	Reserved
13	First Peak/Last Peak Select	R/W	0x0	Retain	This register takes effect only if register 23_5.7:6 = 01. 0 = First Peak 1 = Last Peak
12	Break Link Prior to Measurement	R/W	0x0	Retain	1 = Do not wait 1.5s to break link before starting VCT 0 = Wait 1.5s to break link before starting VCT
11:10	Transmit Pulse Width	R/W	0x11	Retain	00 = full pulse (128ns) 01 = 3/4 pulse 10 = 1/2 pulse 11 = 1/4 pulse
9:8	Transmit Amplitude	R/W	0x0	Retain	00 = full amplitude 01 = 3/4 amplitude 10 = 1/2 amplitude 11 = 1/4 amplitude
7	Distance Measurement Point	R/W	0x0	Retain	If 23_5.7:6 = 00 then 0 = Measure distance when amplitude drops to 50% of peak amplitude 1 = Measure distance at actual maximum amplitude If 23_5.7:6 = 01 then 0 = Measure distance when amplitude drops below hysteresis 1 = Measure distance at actual maximum amplitude If 23_5.7:6 = 1X then this bit is ignored.
6:0	Same-Pair Positive Threshold > 140m	R/W	0x06	Retain	0x00 = 0 mV, 0x01 = 7.81 mV, ..., 0x7F = 992 mV

Table 362: Copper Port Packet Generation
Page 6, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
15:8	Packet Burst	R/W	0x00	Retain	0x00 = Continuous 0x01 to 0xFF = Burst 1 to 255 packets
7	Packet Generator Transmit Trigger	R/W	0x0	Retain	This bit is only valid when all of the following are true: bit 6 =1 bit3 =1 bit15:8 is not equal to all 0s A read of this bit gives the following: 1: Packet generator transmit done 0: Packet generator is transmitting data When this bit is 1 a write of 0 will trigger the packet generator to transmit again. When this bit is 0 a write of 0 or 1 will have no effect.
6	Packet Generator Enable Self Clear Control	R/W	0x0	Retain	0 = Bit 3 will self clear after all packets are sent 1 = Bit 3 will stay high after all packets are sent



Link Street®

88E6321/88E6320 Functional Specification

Table 362: Copper Port Packet Generation (Continued)

Page 6, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
5	Reserved	R/W	0x0	Retain	Reserved
4	Enable CRC checker	R/W	0x0	Retain	1 = Enable 0 = Disable
3	Enable packet generator	R/W	0x0	Retain	1 = Enable 0 = Disable
2	Payload of packet to transmit	R/W	0x0	Retain	0 = Pseudo-random 1 = For 10/100/1000 Mbps = A5, 5A, A5, 5A
1	Length of packet to transmit	R/W	0x0	Retain	1 = 1518 bytes 0 = 64 bytes
0	Transmit an Errorred packet	R/W	0x0	Retain	1 = Tx packets with CRC errors & Symbol Error 0 = No error

Table 363: Copper Port CRC Counters

Page 6, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
15:8	Packet Count	RO	0x00	Retain	0x00 = no packets received 0xFF = 256 packets received (max count). Bit 16_6.4 must be set to 1 in order for register to be valid.
7:0	CRC Error Count	RO	0x00	Retain	0x00=noCRCerrorsdetectedinthe packets received. 0xFF = 256 CRC errors detected in the packets received (max count). Bit 16_6.4 must be set to 1 in order for register to be valid.

Table 364: Checker Control

Page 6, Register 18

Bits	Field	Mode	HW Rst	SW Rst	Description
15:5	Reserved	RO	0x000	Retain	Reserved
4	CRC Counter Reset	R/W, SC	0x0	0x0	1 = Reset This bit will self-clear after write to 1.
3	Enable Stub Test	R/W	0x0	Retain	1 = Enable stub test 0 = Normal Operation
2:0	Reserved	R/W	0x0	Retain	Reserved

Table 365: Misc Test
Page 6, Register 26

Bits	Field	Mode	HW Rst	SW Rst	Description
15	TX_TCLK Enable	R/W	0x0	Retain	The highest numbered enabled port will drive the transmit clock to the HSDACP/N pin. 1 = Enable 0 = Disable
14:13	Reserved	R/W	0x0	Retain	Reserved
12:8	Temperature Threshold	R/W	0x19	Retain	Temperature in C = $5 \times 26_6.4:0 - 25$ i.e. for 100C the value is 11001
7	Temperature Sensor Interrupt Enable	R/W	0x0	Retain	1 = Interrupt Enable 0 = Interrupt Disable
6	Temperature Sensor Interrupt	RO, LH	0x0	0x0	1 = Temperature Reached Threshold 0 = Temperature Below Threshold
5	Reserved	R/W	0x0	Retain	Reserved
4:0	Temperature Sensor (5-bit)	RO	xxxxx	xxxxx	Temperature in C = $5 \times 26_6.4:0 - 25$ i.e. for 100C the value is 11001

Table 366: Temperature Sensor
Page 6, Register 27

Bits	Field	Mode	HW Rst	SW Rst	Description
15:13	Reserved	R/W	0	Retain	
12:11	Temperature Sensor Number of Samples to Average	R/W	0x1	Retain	00 = Average over 5×2^9 samples 01 = Average over 5×2^{11} samples 10 = Average over 5×2^{13} samples 11 = Average over 5×2^{15} samples
10:8	Temperature Sensor Sampling Rate	R/W	0x4	Retain	Sampling Rate 000 = Reserved 001 = Reserved 010 = 168 us 011 = 280 us 100 = 816 us 101 = 2.28 ms 110 = 6.22 ms 111 = 11.79 ms
7:0	Temperature Sensor Alternative Reading (8-bit)	RO	x	x	Temperature in C = $1 \times 27_6.7:0 - 25$. i.e. for 100C the value is 0111_1101



Table 367: PHY Cable Diagnostics Pair 0 Length
Page 7, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Pair 0 Cable Length	RO	0x0000	Retain	Length to fault in meters or centimeters based on register 21_7.10.

Table 368: PHY Cable Diagnostics Pair 1 Length
Page 7, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Pair 1 Cable Length	RO	0x0000	Retain	Length to fault in meters or centimeters based on register 21_7.10.

Table 369: PHY Cable Diagnostics Pair 2 Length
Page 7, Register 18

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Pair 2 Cable Length	RO	0x0000	Retain	Length to fault in meters or centimeters based on register 21_7.10.

Table 370: PHY Cable Diagnostics Pair 3 Length
Page 7, Register 19

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Pair 3 Cable Length	RO	0x0000	Retain	Length to fault in meters or centimeters based on register 21_7.10.

Table 371: PHY Cable Diagnostics Results
Page 7, Register 20

Bits	Field	Mode	HW Rst	SW Rst	Description
15:12	Pair 3 Fault Code	RO	0x0	Retain	0000 = Invalid 0001 = Pair Ok 0010 = Pair Open 0011 = Same Pair Short 0100 = Cross Pair Short 1001 = Pair Busy Else = Reserved
11:8	Pair 2 Fault Code	RO	0x0	Retain	0000 = Invalid 0001 = Pair Ok 0010 = Pair Open 0011 = Same Pair Short 0100 = Cross Pair Short 1001 = Pair Busy Else = Reserved
7:4	Pair 1 Fault Code	RO	0x0	Retain	0000 = Invalid 0001 = Pair Ok 0010 = Pair Open 0011 = Same Pair Short 0100 = Cross Pair Short 1001 = Pair Busy Else = Reserved
3:0	Pair 0 Fault Code	RO	0x0	Retain	0000 = Invalid 0001 = Pair Ok 0010 = Pair Open 0011 = Same Pair Short 0100 = Cross Pair Short 1001 = Pair Busy Else = Reserved



Link Street®

88E6321/88E6320 Functional Specification

Table 372: PHY Cable Diagnostics Control
Page 7, Register 21

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Run Immediately	R/W, SC	0x0	0x0	0 = No Action 1 = Run VCT™ Now
14	Run At Each Auto-Negotiation Cycle	R/W	0x1	Retain	0 = Do Not Run At Auto-Negotiation Cycle 1 = Run At Auto-Negotiation Cycle
13	Disable Cross Pair Check	R/W	0x0	Retain	0 = Enable Cross Pair Check 1 = Disable Cross Pair Check
12	Run After Break Link	R/W, SC	0x0	0x0	0 = No Action 1 = Run VCT After Breaking Link
11	Cable Diagnostics Status	RO	0x0	Retain	0 = Complete 1 = In Progress
10	Cable Length Unit	R/W	0x0	Retain	0 = Centimeters 1 = Meters
9:0	Reserved	RO	0x000	0x000	Reserved

Table 373: Advanced VCT™ Cross Pair Negative Threshold
Page 7, Register 25

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14:8	Cross Pair Negative Threshold > 30m	R/W	0x01	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV
7	Reserved	RO	0x0	0x0	Reserved
6:0	Cross Pair Negative Threshold < 30m	R/W	0x04	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV

Table 374: Advanced VCT Same Pair Impedance Negative Threshold 0 and 1
Page 7, Register 26

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14:8	Same-Pair Negative Threshold 10m - 50m	R/W	0x0F	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV
7	Reserved	RO	0x0	0x0	Reserved
6:0	Same-Pair Negative Threshold < 10m	R/W	0x12	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV

Table 375: Advanced VCT Same Pair Impedance Negative Threshold 2 and 3
Page 7, Register 27

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14:8	Same-Pair Negative Threshold 110m - 140m	R/W	0x0A	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV
7	Reserved	RO	0x0	0x0	Reserved
6:0	Same-Pair Negative Threshold 50m - 110m	R/W	0x0C	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV

Table 376: Advanced VCT Same Pair Impedance Negative Threshold 4
Page 7, Register 28

Bits	Field	Mode	HW Rst	SW Rst	Description
15:7	Reserved	RO	0x000	0x000	Reserved
6:0	Same-Pair Negative Threshold > 140m	R/W	0x06	Retain	0x00 = 0 mV, 0x01 = - 7.81 mV, ..., 0x7F = - 992 mV



Link Street®

88E6321/88E6320 Functional Specification

11.1.2 XMDIO Register Description

Table 377: Register Map

Register Name	Register Address	Table and Page
PCS control 1 register	Device 3, Register 0	Table 378, p. 489
PCS status 1 register	Device 3, Register 1	Table 379, p. 489
PCS EEE capability register	Device 3, Register 20	Table 380, p. 490
PCS EEE wake error counter	Device 3, Register 22	Table 381, p. 490
EEE advertisement register	Device 7, Register 60	Table 382, p. 491
EEE Link partner advertisement register	Device 7, Register 61	Table 383, p. 491

Table 378: PCS control 1 register
Device 3, Register 0

Bits	Field	Mode	HW Rst	SW Rst	Description
15:11	Reserved	RO	0x00	Retain	Reserved
10	Clock Stopable	R/W	0x0	Retain	1 = Clock stopable during LPI 0 = Clock not stoppable
9:0	Reserved	RO	0x000	Retain	Reserved

Table 379: PCS status 1 register
Device 3, Register 1

Bits	Field	Mode	HW Rst	SW Rst	Description
15:12	Reserved	RO	0x0	Retain	Reserved
11	Tx LP idle received	RO/LH	0x0	Retain	1 = Tx PCS has received LP idle 0 = LP Idle not received
10	Rx LP idle received	RO/LH	0x0	Retain	1 = Rx PCS has received LP idle 0 = LP Idle not received
9	Tx LP idle indication	RO	0x0	Retain	1 = Tx PCS is currently receiving LP idle 0 = PCS is not currently receiving LP idle
8	Rx LP idle indication	RO	0x0	Retain	1 = Rx PCS is currently receiving LP idle 0 = PCS is not currently receiving LP idle
7:3	Reserved	RO	0x08	Retain	Reserved
2	PCS receive link status	RO	0x0	Retain	1 = PCS receive link up 0 = PCS receive link down
1	Low-power ability	RO	0x1	Retain	1 = PCS supports low-power mode 0 = PCS does not support low-power mode
0	Reserved	RO	0x0	Retain	Reserved



Table 380: PCS EEE capability register
Device 3, Register 20

Bits	Field	Mode	HW Rst	SW Rst	Description
15:7	Reserved	RO	0x000	Retain	Reserved
6	10GBASE-KR EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-KR 0 = EEE is not supported for 10GBASE-KR
5	10GBASE-KX4 EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-KX4 0 = EEE is not supported for 1000BASE-KX RO
4	1000BASE-KX	RO	0x0	Retain	1 = EEE is supported for 1000BASE-KX 0 = EEE is not supported for 1000BASE-KX RO
3	10GBASE-T EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-T 0 = EEE is not supported for 10GBASE-T
2	1000BASE-T EEE	RO	0x1	Retain	1 = EEE is supported for 1000BASE-T 0 = EEE is not supported for 1000BASE-T
1	100BASE-TX EEE	RO	0x1	Retain	1 = EEE is supported for 100BASE-TX 0 = EEE is not supported for 100BASE-TX
0	Reserved	RO	0x0	Retain	Reserved

Table 381: PCS EEE wake error counter
Device 3, Register 22

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	EEE wake error counter	ROC	0x0000	Retain	This counter is incremented for each transition of lpi_wake_timer_done from FALSE to TRUE

Table 382: EEE advertisement register
Device 7, Register 60

Bits	Field	Mode	HW Rst	SW Rst	Description
15:7	Reserved	RO	0x000	Retain	Reserved
6	10GBASE-KR EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-KR 0 = EEE is not supported for 10GBASE-KR
5	10GBASE-KX4 EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-KX4 0 = EEE is not supported for 1000BASE-KX RO
4	1000BASE-KX	RO	0x0	Retain	1 = EEE is supported for 1000BASE-KX 0 = EEE is not supported for 1000BASE-KX RO
3	10GBASE-T EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-T 0 = EEE is not supported for 10GBASE-T
2	1000BASE-T EEE	R/W	0x0	Retain	Upon hardware reset this bit takes on the value of pg_config_eee_1000_a. When pd_aneg_now_a is asserted this bit takes on the value of pg_config_eee_1000_a. 1 = EEE is supported for 1000BASE-T 0 = EEE is not supported for 1000BASE-T
1	100BASE-TX EEE	R/W	0x0	Retain	Upon hardware reset this bit takes on the value of pg_config_eee_100_a. When pd_aneg_now_a is asserted this bit takes on the value of pg_config_eee_100_a. 1 = EEE is supported for 100BASE-TX 0 = EEE is not supported for 100BASE-TX
0	Reserved	RO	0x0	Retain	Reserved

Table 383: EEE Link partner advertisement register
Device 7, Register 61

Bits	Field	Mode	HW Rst	SW Rst	Description
15:7	Reserved	RO	0x000	Retain	Reserved
6	LP 10GBASE-KR EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-KR 0 = EEE is not supported for 10GBASE-KR
5	LP 10GBASE-KX4 EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-KX4 0 = EEE is not supported for 1000BASE-KX RO
4	LP 1000BASE-KX	RO	0x0	Retain	1 = EEE is supported for 1000BASE-KX 0 = EEE is not supported for 1000BASE-KX RO
3	LP 10GBASE-T EEE	RO	0x0	Retain	1 = EEE is supported for 10GBASE-T 0 = EEE is not supported for 10GBASE-T
2	LP 1000BASE-T EEE	RO	0x0	Retain	1 = EEE is supported for 1000BASE-T 0 = EEE is not supported for 1000BASE-T
1	LP 100BASE-TX EEE	RO	0x0	Retain	1 = EEE is supported for 100BASE-TX 0 = EEE is not supported for 100BASE-TX
0	Reserved	RO	0x0	Retain	Reserved



Link Street®

88E6321/88E6320 Functional Specification

11.2

SERDES Register Description

The device supports both Clause 22 MDIO register access protocol and Clause 45 XMDIO register access protocol. The device also supports Clause 22 MDIO access to registers in Clause 45 XMDIO space using Page 0 register 13 and 14.

**Note**

The SERDES registers in the device are not accessible directly. Instead, these registers must be accessed through the SMI PHY Command and Data Switch Registers in the Global 2 register space (Global 2 Offsets 0x18 and 0x19).

Table 384 below defines the register types used in the register map.

Table 384: Register Types

Type	Description
C	Clear after read.
LH	Register field with latching high function. If status is high, then the register is set to one and remains set until a read operation is performed through the management interface or a reset occurs.
LL	Register field with latching low function. If status is low, then the register is cleared to zero and remains zero until a read operation is performed through the management interface or a reset occurs.
Retain	The register value is retained after software reset is executed.
RES	Reserved for future use. All reserved bits are read as zero unless otherwise noted.
RO	Read only.
ROS	Read only, Set high after read.
ROC	Read only clear. After read, register field is cleared.
R/W	This bit or these bits must be read and left unchanged when performing a write.
RWC	Read/Write clear on read. All field bits are readable and writable. After reset or after the register field is read, register field is cleared to zero.
RWR	Read/Write clear on read. All field bits are readable and writable. After reset, register field is cleared to 0.
RWS	Read/Write set. All field bits are readable and writable. After reset, register field is set to a non-zero value specified in the text.
SC	Self-Clear. Writing a one to this register causes the desired function to be immediately executed, then the register field is automatically cleared to zero when the function is complete.
Update	Value written to the register field doesn't take effect until soft reset is executed.
WO	Write only. Reads to this type of register field return undefined data.

For all binary equations appearing in the register map, the symbol | is equal to a binary OR operation.

Table 385: Fiber/SERDES Register Map Summary - Page 1

		Page Address							
		0	1	2	3	4	5	6	7
Register Address	Fiber/SERDES								
	0	Fiber Control Register							
	1	Fiber Status Register							
	2	PHY Identifier 1							
	3	PHY Identifier 2							
	4	Fiber Auto-Negotiation Advertisement Register							
	5	Fiber Link Partner Ability Register							
	6	Fiber Auto-Negotiation Expansion Register							
	7	Fiber Next Page Transmit Register							
	8	Fiber Link Partner Next Page Register							
	9								
	10								
	11								
	12								
	13								
	14								
	15	Extended Status Register							
	16	Fiber Specific Control Register 1							
	17	Fiber Specific Status Register							
	18	Fiber Specific Interrupt Enable Register							
	19	Fiber Interrupt Status Register							
	20								
	21	Fiber Specific Receive Error Counter							
	22	Page Address							
	23	PRBS Control							
	24	PRBS Error Counter LSB							
	25	PRBS Error Counter MSB							
	26	Fiber Specific Control Register 2							
	27								
	28								
	29								
	30								
	31								



Link Street®

88E6321/88E6320 Functional Specification

Table 386: Fiber/SERDES Registers - Register Map

Register Name	Register Address	Table and Page
Fiber Control Register	Page 1, Register 0	Table 387, p. 495
Fiber Status Register	Page 1, Register 1	Table 388, p. 497
PHY Identifier	Page 1, Register 2	Table 389, p. 498
PHY Identifier	Page 1, Register 3	Table 390, p. 498
Fiber Auto-Negotiation Advertisement Register - 1000BASE-X Mode (Register 16_1:0 = 01)	Page 1, Register 4	Table 391, p. 499
Fiber Auto-Negotiation Advertisement Register - SGMII (System mode) (Register 16_1:0 = 10)	Page 1, Register 4	Table 392, p. 501
Fiber Auto-Negotiation Advertisement Register - SGMII (Media mode) (Register 16_1:0 = 11)	Page 1, Register 4	Table 393, p. 501
Fiber Link Partner Ability Register - 1000BASE-X Mode (Register 16_1:0 = 01)	Page 1, Register 5	Table 394, p. 502
Fiber Link Partner Ability Register - SGMII (System mode) (Register 16_1:0 = 10)	Page 1, Register 5	Table 395, p. 503
Fiber Link Partner Ability Register - SGMII (Media mode) (Register 16_1:0 = 11)	Page 1, Register 5	Table 396, p. 503
Fiber Auto-Negotiation Expansion Register	Page 1, Register 6	Table 397, p. 504
Fiber Next Page Transmit Register	Page 1, Register 7	Table 398, p. 504
Fiber Link Partner Next Page Register	Page 1, Register 8	Table 399, p. 505
Extended Status Register	Page 1, Register 15	Table 400, p. 505
Fiber Specific Control Register 1	Page 1, Register 16	Table 401, p. 505
Fiber Specific Status Register	Page 1, Register 17	Table 402, p. 507
Fiber Interrupt Enable Register	Page 1, Register 18	Table 403, p. 508
Fiber Interrupt Status Register	Page 1, Register 19	Table 404, p. 509
Fiber Receive Error Counter Register	Page 1, Register 21	Table 405, p. 509
Fiber Page Register	Page 1, Register 22	Table 406, p. 509
PRBS Control	Page 1, Register 23	Table 407, p. 510
PRBS Error Counter LSB	Page 1, Register 24	Table 408, p. 510
PRBS Error Counter MSB	Page 1, Register 25	Table 409, p. 510
Fiber Specific Control Register 2	Page 1, Register 26	Table 410, p. 511

Table 387: Fiber Control Register
Page 1, Register 0

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Fiber Reset	R/W	0x0	SC	Fiber Software Reset. Affects page 1. Writing a 1 to this bit causes the PHY state machines to be reset. When the reset operation is done, this bit is cleared to 0 automatically. The reset occurs immediately. 1 = PHY reset 0 = Normal operation
14	Loopback	R/W	0x0	0x0	When loopback is activated, the transmitter data presented on TXD of the internal bus is looped back to RXD of the internal bus. Link is broken when loopback is enabled. Loopback speed is determined by the mode the device is in. 1000BASE-X - loopback is always in 1000Mbps. 100BASE-FX - loopback is always in 100Mbps. 1 = Enable Loopback 0 = Disable Loopback
13	Speed Select (LSB)	RO, R/W	0x0	Retain	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is always 1. If register 16_1.1:0 (MODE[1:0]) = 01 then this bit is always 0. If register 16_1.1:0 (MODE[1:0]) = 10 then this bit is 1 when the PHY is at 100 Mbps, else it is 0. If register 16_1.1:0 (MODE[1:0]) = 11 then this bit is R/W. bit 6,13 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps
12	Auto-Negotiation Enable	R/W	See Descr	Retain	If the value of this bit is changed, the link will be broken and Auto-Negotiation Restarted This bit has no effect when in 100BASE-FX mode When this bit gets set/reset, Auto-negotiation is restarted (bit 0_1.9 is set to 1). On hardware reset this bit takes on the value of S_ANEG 1 = Enable Auto-Negotiation Process 0 = Disable Auto-Negotiation Process
11	Power Down	R/W	See Descr	0x0	When the port is switched from power down to normal operation, software reset and restart Auto-Negotiation are performed even when bits Reset (0_1.15) and Restart Auto-Negotiation (0_1.9) are not set by the user. On hardware reset, bit 0_1.11 takes on the value of PDOWN and (MODE[2:0] = 01x or 11x or 100) 1 = Power down 0 = Normal operation
10	Isolate	RO	0x0	0x0	This function is not supported
9	Restart Fiber Auto-Negotiation	R/W, SC	0x0	SC	Auto-Negotiation automatically restarts after hardware, software reset (0_1.15) or change in auto-negotiation enable (0_1.12) regardless of whether or not the restart bit (0_1.9) is set. The bit is set when Auto-negotiation is Enabled or Disabled in 0_1.12 1 = Restart Auto-Negotiation Process 0 = Normal operation



Link Street®

88E6321/88E6320 Functional Specification

Table 387: Fiber Control Register (Continued)

Page 1, Register 0

Bits	Field	Mode	HW Rst	SW Rst	Description
8	Duplex Mode	R/W	0x1	Retain	Writing this bit has no effect unless one of the following events occur: Software reset is asserted (Register 0_1.15) Restart Auto-Negotiation is asserted (Register 0_1.9) Auto-Negotiation Enable changes (Register 0_1.12) Power down (Register 0_1.11) transitions from power down to normal operation 1 = Full-duplex 0 = Half-duplex
7	Collision Test	RO	0x0	0x0	This bit has no effect.
6	Speed Selection (MSB)	RO, R/W	0x1	Retain	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is always 0. If register 16_1.1:0 (MODE[1:0]) = 01 then this bit is always 1. If register 16_1.1:0 (MODE[1:0]) = 10 then this bit is 1 when the PHY is at 1000 Mbps, else it is 0. If register 16_1.1:0 (MODE[1:0]) = 11 then this bit is R/W. bit 6,13 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps
5:0	Reserved	RO	Always 000000	Always 000000	Reserved

Table 388: Fiber Status Register
Page 1, Register 1

Bits	Field	Mode	HW Rst	SW Rst	Description
15	100BASE-T4	RO	Always 0	Always 0	100BASE-T4. This protocol is not available. 0 = PHY not able to perform 100BASE-T4
14	100BASE-X Full-Duplex	RO	See Descr	See Descr	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is 1, else this bit is 0. bit 6,13 1 = PHY able to perform full duplex 100BASE-X 0 = PHY not able to perform full duplex 100BASE-X
13	100BASE-X Half-Duplex	RO	See Descr	See Descr	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is 1, else this bit is 0. bit 6,13 1 = PHY able to perform half-duplex 100BASE-X 0 = PHY not able to perform half-duplex 100BASE-X
12	10 Mbps Full Duplex	RO	Always 0	Always 0	0 = PHY not able to perform full-duplex 10BASE-T
11	10 Mbps Half-Duplex	RO	Always 0	Always 0	0 = PHY not able to perform half-duplex 10BASE-T
10	100BASE-T2 Full-Duplex	RO	Always 0	Always 0	This protocol is not available. 0 = PHY not able to perform full-duplex
9	100BASE-T2 Half-Duplex	RO	Always 0	Always 0	This protocol is not available. 0 = PHY not able to perform half-duplex
8	Extended Status	RO	Always 1	Always 1	1 = Extended status information in Register 15
7	Reserved	RO	Always 0	Always 0	Reserved
6	MF Preamble Suppression	RO	Always 1	Always 1	1 = PHY accepts management frames with preamble suppressed
5	Fiber Auto-Negotiation Complete	RO	0x0	0x0	1 = Auto-Negotiation process complete 0 = Auto-Negotiation process not complete Bit is not set when link is up due of Fiber Auto-negotiation Bypass or if Auto-negotiation is disabled.
4	Fiber Remote Fault	RO,LH	0x0	0x0	1 = Remote fault condition detected 0 = Remote fault condition not detected This bit is always 0 in SGMII modes.
3	Auto-Negotiation Ability	RO	See Descr	See Descr	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is 0, else this bit is 1. bit 6,13 1 = PHY able to perform Auto-Negotiation 0 = PHY not able to perform Auto-Negotiation
2	Fiber Link Status	RO,LL	0x0	0x0	This register bit indicates when link was lost since the last read. For the current link status, either read this register back-to-back or read Register 17_1.10 Link Real Time. 1 = Link is up 0 = Link is down
1	Reserved	RO,LH	Always 0	Always 0	Reserved
0	Extended Capability	RO	Always 1	Always 1	1 = Extended register capabilities



Table 389: PHY Identifier
Page 1, Register 2

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Organizationally Unique Identifier Bit 3:18	RO	0x0141	0x0141	<p>Marvell® OUI is 0x005043 0000 0000 0101 0000 0100 0011 ^ ^ bit 1.....bit 24</p> <p>Register 2.[15:0] show bits 3 to 18 of the OUI. 0000000101000001 ^ ^ bit 3.....bit18</p>

Table 390: PHY Identifier
Page 1, Register 3

Bits	Field	Mode	HW Rst	SW Rst	Description
15:10	OUI Lsb	RO	Always 000011	Always 000011	Organizationally Unique Identifier bits 19:24 000011 ^^ bit 19...bit24
9:4	Model Number	RO	See Descr	See Descr	Contact Marvell® FAEs for information on the device model number.
3:0	Revision Number	RO	See Descr	See Descr	Rev Number See relevant product Release Notes for details.

Table 391: Fiber Auto-Negotiation Advertisement Register - 1000BASE-X Mode (Register 16_1.1:0 = 01)
Page 1, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	R/W	0x0	Retain	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (Register 0_1.15) Restart Auto-Negotiation is asserted (Register 0_1.9) Power down (Register 0_1.11) transitions from power down to normal operation Link goes down 1 = Advertise 0 = Not advertised
14	Reserved	RO	Always 0	Always 0	Reserved
13:12	Remote Fault 2/ RemoteFault 1	R/W	0x0	Retain	A write to this register bit does not take effect until any one of the following also occurs: Software reset is asserted (Register 0_1.15) Re-start Auto-Negotiation is asserted (Register 0_1.9) Power down (Register 0_1.11) transitions from power down to normal operation Link goes down Device has no ability to detect remote fault. 00 = No error, link OK (default) 01 = Link Failure 10 = Offline 11 = Auto-Negotiation Error
11:9	Reserved	RO	Always 000	Always 000	Reserved
8:7	Pause	R/W	See Descr.	Retain	A write to this register bit does not take effect until any one of the following also occurs: Software reset is asserted (Register 0_1.15) Re-start Auto-Negotiation is asserted (Register 0_1.9) Power down (Register 0_1.11) transitions from power down to normal operation Link goes down Upon hardware reset both bits takes on the value of ENA_PAUSE. 00 = No PAUSE 01 = Symmetric PAUSE 10 = Asymmetric PAUSE toward link partner 11 = Both Symmetric PAUSE and Asymmetric PAUSE toward local device.
6	1000BASE-X Half-Duplex	R/W	See Descr.	Retain	A write to this register bit does not take effect until any one of the following also occurs: Software reset is asserted (Register 0_1.15) Re-start Auto-Negotiation is asserted (Register 0_1.9) Power down (Register 0_1.11) transitions from power down to normal operation Link goes down Upon hardware reset this bit takes on the value of C_ANEG[0]. 1 = Advertise 0 = Not advertised



Link Street®

88E6321/88E6320 Functional Specification

Table 391: Fiber Auto-Negotiation Advertisement Register - 1000BASE-X Mode (Register 16_1.1:0 = 01) (Continued)
Page 1, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
5	1000BASE-X Full-Duplex	R/W	0x1	Retain	A write to this register bit does not take effect until any one of the following also occurs: Software reset is asserted (Register 0_1.15) Re-start Auto-Negotiation is asserted (Register 0_1.9) Power down (Register 0_1.11) transitions from power down to normal operation Link goes down 1 = Advertise 0 = Not advertised
4:0	Reserved	R/W	0x00	0x00	Reserved

Table 392: Fiber Auto-Negotiation Advertisement Register - SGMII (System mode) (Register 16_1.1:0 = 10)
Page 1, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Link Status	RO	0x0	0x0	0 = Link is Not up on the Copper Interface 1 = Link is up on the Copper Interface
14	Reserved	RO	Always 0	Always 0	Reserved
13	Reserved	RO	Always 0	Always 0	Reserved
12	Duplex Status	RO	0x0	0x0	0 = Interface Resolved to Half-duplex 1 = Interface Resolved to Full-duplex
11:10	Speed[1:0]	RO	0x0	0x0	00 = Interface speed is 10 Mbps 01 = Interface speed is 100 Mbps 10 = Interface speed is 1000 Mbps 11 = Reserved
9	Transmit Pause	RO	0x0	0x0	Note that if register 16_1.7 is set to 0 then this bit is always forced to 0. 0 = Disabled, 1 = Enabled
8	Receive Pause	RO	0x0	0x0	Note that if register 16_1.7 is set to 0 then this bit is always forced to 0. 0 = Disabled, 1 = Enabled
7	Fiber/Copper	RO	0x0	0x0	Note that if register 16_1.7 is set to 0 then this bit is always forced to 0. 0 = Copper media, 1 = Fiber media
6:0	Reserved	RO	Always 0000001	Always 0000001	Reserved

Table 393: Fiber Auto-Negotiation Advertisement Register - SGMII (Media mode) (Register 16_1.1:0 = 11)
Page 1, Register 4

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Reserved	RO	Always 0x0001	Always 0x0001	Reserved



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 394: Fiber Link Partner Ability Register - 1000BASE-X Mode (Register 16_1.1:0 = 01)
Page 1, Register 5

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word Bit 15 1 = Link partner capable of next page 0 = Link partner not capable of next page
14	Acknowledge	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Acknowledge Received Code Word Bit 14 1 = Link partner received link code word 0 = Link partner has not received link code word
13:12	Remote Fault 2/ Remote Fault 1	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word Bit 13:12 00 = No error, link OK (default) 01 = Link Failure 10 = Offline 11 = Auto-Negotiation Error
11:9	Reserved	RO	0x0	0x0	Reserved
8:7	Asymmetric Pause	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word Bit 8:7 00 = No PAUSE 01 = Symmetric PAUSE 10 = Asymmetric PAUSE toward link partner 11 = Both Symmetric PAUSE and Asymmetric PAUSE toward local device.
6	1000BASE-X Half-Duplex	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word bit 6 1 = Link partner capable of 1000BASE-X half-duplex. 0 = Link partner not capable of 1000BASE-X half-duplex.
5	1000BASE-X Full-Duplex	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word bit 5 1 = Link partner capable of 1000BASE-X full-duplex. 0 = Link partner not capable of 1000BASE-X full-duplex.
4:0	Reserved	RO	0x00	0x00	Reserved

Table 395: Fiber Link Partner Ability Register - SGMII (System mode) (Register 16_1.1:0 = 10)
Page 1, Register 5

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	0x0	0x0	Reserved
14	Acknowledge	RO	0x0	0x0	Acknowledge Register bit is cleared when link goes down and loaded when a base page is received Received Code Word Bit 14 1 = Link partner received link code word 0 = Link partner has not received link code word
13:0	Reserved	RO	0x0000	0x0000	Reserved

Table 396: Fiber Link Partner Ability Register - SGMII (Media mode) (Register 16_1.1:0 = 11)
Page 1, Register 5

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Link	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word Bit 15 1 = Copper Link is up on the link partner 0 = Copper Link is not up on the link partner
14	Acknowledge	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Acknowledge Received Code Word Bit 14 1 = Link partner received link code word 0 = Link partner has not received link code word
13	Reserved	RO	0x0	0x0	Reserved
12	Duplex Status	RO	0x0	0x0	Register bit is cleared when link goes down and loaded when a base page is received Received Code Word Bit 12 1 = Copper Interface on the link Partner is capable of Full-duplex 0 = Copper Interface on the link partner is capable of Half-duplex
11:10	Speed Status	RO	0x0	0x0	Register bits are cleared when link goes down and loaded when a base page is received Received Code Word Bit 11:10 00 = 10 Mbps 01 = 100 Mbps 10 = 1000 Mbps 11 = Reserved
9	Transmit Pause Status	RO	0x0	0x0	This bit is non-zero only if the link partner supports enhanced SGMII auto negotiation. Received Code Word Bit 9 0 = Disabled, 1 = Enabled
8	Receive Pause Status	RO	0x0	0x0	This bit is non-zero only if the link partner supports enhanced SGMII auto negotiation. Received Code Word Bit 8 0 = Disabled, 1 = Enabled



Link Street®

88E6321/88E6320 Functional Specification

**Table 396: Fiber Link Partner Ability Register - SGMII (Media mode) (Register 16_1.1:0 = 11)
(Continued)**

Bits	Field	Mode	HW Rst	SW Rst	Description
7	Fiber/Copper Status	RO	0x0	0x0	This bit is non-zero only if the link partner supports enhanced SGMII auto negotiation. Received Code Word Bit 7 0 = Copper media, 1 = Fiber media
6:0	Reserved	RO	0x00	0x00	Reserved

**Table 397: Fiber Auto-Negotiation Expansion Register
Page 1, Register 6**

Bits	Field	Mode	HW Rst	SW Rst	Description
15:4	Reserved	RO	0x000	0x000	Reserved
3	Link Partner Next page Able	RO	0x0	0x0	SGMII and 100BASE-FX modes this bit is always 0. In 1000BASE-X mode register 6_1.3 is set when a base page is received and the received link control word has bit 15 set to 1. The bit is cleared when link goes down. 1 = Link Partner is Next Page able 0 = Link Partner is not Next Page able
2	Local Next Page Able	RO	Always 1	Always 1	1 = Local Device is Next Page able
1	Page Received	RO, LH	0x0	0x0	Register 6_1.1 is set when a valid page is received. 1 = A New Page has been received 0 = A New Page has not been received
0	Link Partner Auto-Negotiation Able	RO	0x0	0x0	This bit is set when there is sync status, the fiber receiver has received 3 non-zero matching valid configuration code groups and Auto-negotiation is enabled in register 0_1.12 1 = Link Partner is Auto-Negotiation able 0 = Link Partner is not Auto-Negotiation able

**Table 398: Fiber Next Page Transmit Register
Page 1, Register 7**

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	R/W	0x0	0x0	A write to register 7_1 implicitly sets a variable in the Auto-Negotiation state machine indicating that the next page has been loaded. Register 7_1 only has effect in the 1000BASE-X mode. Transmit Code Word Bit 15
14	Reserved	RO	0x0	0x0	Transmit Code Word Bit 14
13	Message Page Mode	R/W	0x1	0x1	Transmit Code Word Bit 13
12	Acknowledge2	R/W	0x0	0x0	Transmit Code Word Bit 12
11	Toggle	RO	0x0	0x0	Transmit Code Word Bit 11. This bit is internally set to the opposite value each time a page is received
10:0	Message/ Unformatted Field	R/W	0x001	0x001	Transmit Code Word Bit 10:0

Table 399: Fiber Link Partner Next Page Register
Page 1, Register 8

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Next Page	RO	0x0	0x0	Register 8_1 only has effect in the 1000BASE-X mode. The register is loaded only when a next page is received from the link partner. It is cleared each time the link goes down. Received Code Word Bit 15
14	Acknowledge	RO	0x0	0x0	Received Code Word Bit 14
13	Message Page	RO	0x0	0x0	Received Code Word Bit 13
12	Acknowledge2	RO	0x0	0x0	Received Code Word Bit 12
11	Toggle	RO	0x0	0x0	Received Code Word Bit 11
10:0	Message/ Unformatted Field	RO	0x000	0x000	Received Code Word Bit 10:0

Table 400: Extended Status Register
Page 1, Register 15

Bits	Field	Mode	HW Rst	SW Rst	Description
15	1000BASE-X Full-Duplex	RO	See Descr	See Descr	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is 0, else this bit is 1. 1 = 1000BASE-X full duplex capable 0 = not 1000BASE-X full duplex capable
14	1000BASE-X Half-Duplex	RO	See Descr	See Descr	If register 16_1.1:0 (MODE[1:0]) = 00 then this bit is 0, else this bit is 1. 1 = 1000BASE-X half duplex capable 0 = not 1000BASE-X half duplex capable
13	1000BASE-T Full-Duplex	RO	0x0	0x0	0 = not 1000BASE-T full duplex capable
12	1000BASE-T Half-Duplex	RO	0x0	0x0	0 = not 1000BASE-T half duplex capable
11:0	Reserved	RO	0x000	0x000	Reserved

Table 401: Fiber Specific Control Register 1
Page 1, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Fiber Transmit FIFO Depth	R/W	0x1	Retain	00 = ± 16 Bits 01 = ± 24 Bits 10 = ± 32 Bits 11 = ± 40 Bits
13	Block Carrier Extension Bit	R/W	0x0	Retain	Carrier extension and carrier extension with error are converted to idle symbols on the RXD only during full duplex mode. 1 = Enable Block Carrier Extension 0 = Disable Block Carrier Extension



Link Street®

88E6321/88E6320 Functional Specification

MARVELL CONFIDENTIAL - UNAUTHORIZED DISTRIBUTION OR USE STRICTLY PROHIBITED

Table 401: Fiber Specific Control Register 1

Page 1, Register 16

Bits	Field	Mode	HW Rst	SW Rst	Description
12	SERDES Loop-back	R/W	0x0	0x0	Register 16_1.8 selects the line loopback path. 1 = Enable loopback from SERDES input to SERDES output 0 = Normal Operation
11	Assert CRS on Transmit	R/W	0x0	Retain	This bit has no effect in full-duplex. 1 = Assert on transmit 0 = Never assert on transmit
10	Force Link Good	R/W	0x0	Retain	If link is forced to be good, the link state machine is bypassed and the link is always up. 1 = Force link good 0 = Normal operation
9	Reserved	R/W	0x0	Retain	Reserved
8	SERDES Loop-back Type	R/W	0x0	Retain	0 = Loopback Through PCS (Tx and Rx can be asynchronous) 1 = Loopback raw 10 bit data (Tx and Rx must be synchronous)
7:6	Enhanced SGMII	R/W	0x1	Update	00 = Do not pass Flow Control through SGMII Auto-negotiation 01 = Pass Flow Control through SGMII Auto-negotiation (MODE 1 – Marvell Legacy Mode) 10 = Pass Flow Control through SGMII Auto-negotiation (MODE 2) 11 = Reserved
5:4	Reserved	R/W	0x0	Retain	Reserved
3	MAC Interface Power Down	R/W	0x1	Update	Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect. This bit determines whether MAC interface powers down when Register 0_1.11 is used to power down the device or when the PHY enters the energy detect state. 1 = Always power up 0 = Can power down
2	Reserved	R/W	0x1	Retain	Must set to 1
1:0	MODE[1:0]	RO	See Desc.	See Desc.	These bits reflects the mode as programmed in register of 20_18.2:0 00 = 100BASE-FX 01 = 1000BASE-X 10 = SGMII System mode 11 = SGMII Media mode

Table 402: Fiber Specific Status Register
Page 1, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
15:14	Speed	RO	0x0	Retain	<p>These status bits are valid only after resolved bit 17_1.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. In 100BASE-FX mode this bit is always 01.</p> <p>11 = Reserved 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps</p>
13	Duplex	RO	0x0	Retain	<p>This status bit is valid only after resolved bit 17_1.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. In 100BASE-FX mode this bit follows register 0_1.8.</p> <p>1 = Full-duplex 0 = Half-duplex</p>
12	Page Received	RO, LH	0x0	0x0	<p>In 100BASE-FX mode this bit is always 0.</p> <p>1 = Page received 0 = Page not received</p>
11	Speed and Duplex Resolved	RO	0x0	0x0	<p>When Auto-Negotiation is not enabled or in 100BASE-FX mode this bit is always 1.</p> <p>1 = Resolved 0 = Not resolved</p> <p>If bit 26_1.5 is 1, then this bit will be 0.</p>
10	Link (real time)	RO	0x0	0x0	<p>1 = Link up 0 = Link down</p>
9:6	Reserved	RO	Always 00000	Always 00000	Reserved
5	Sync status	RO	0x0	0x0	<p>1 = Sync 0 = No Sync</p>
4	Fiber Energy Detect Status	RO	0x1	0x1	<p>1 = No energy detected 0 = Energy Detected</p>
3	Transmit Pause Enabled	RO	0x0	0x0	<p>This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the device. This status bit is valid only after resolved bit 17_1.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. In 100BASE-FX mode this bit is always 0.</p> <p>1 = Transmit pause enabled 0 = Transmit pause disable</p>
2	Receive Pause Enabled	RO	0x0	0x0	<p>This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the device. This status bit is valid only after resolved bit 17_1.11 = 1. The resolved bit is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. In 100BASE-FX mode this bit is always 0.</p> <p>1 = Receive pause enabled 0 = Receive pause disabled</p>



Link Street®

88E6321/88E6320 Functional Specification

Table 402: Fiber Specific Status Register (Continued)

Page 1, Register 17

Bits	Field	Mode	HW Rst	SW Rst	Description
1:0	Reserved	RO	Always 00	Always 00	Reserved

Table 403: Fiber Interrupt Enable Register

Page 1, Register 18

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	Always 0	Always 0	Reserved
14	Speed Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
13	Duplex Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
12	Page Received Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
11	Auto-Negotiation Completed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
10	Link Status Changed Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
9	Symbol Error Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
8	False Carrier Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
7	FIFO Over/Underflow Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
6:5	Reserved	RO	Always 00	Always 00	Reserved
4	Fiber Energy Detect Interrupt Enable	R/W	0x0	Retain	1 = Interrupt enable 0 = Interrupt disable
3:0	Reserved	RO	Always 0000	Always 0000	Reserved

Table 404: Fiber Interrupt Status Register
Page 1, Register 19

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reserved	RO	Always 0	Always 0	Reserved
14	Speed Changed	RO,LH	0x0	0x0	1 = Speed changed 0 = Speed not changed
13	Duplex Changed	RO,LH	0x0	0x0	1 = Duplex changed 0 = Duplex not changed
12	Page Received	RO,LH	0x0	0x0	1 = Page received 0 = Page not received
11	Auto-Negotiation Completed	RO,LH	0x0	0x0	1 = Auto-Negotiation completed 0 = Auto-Negotiation not completed
10	Link Status Changed	RO,LH	0x0	0x0	1 = Link status changed 0 = Link status not changed
9	Symbol Error	RO,LH	0x0	0x0	1 = Symbol error 0 = No symbol error
8	False Carrier	RO,LH	0x0	0x0	1 = False carrier 0 = No false carrier
7	FIFO Over/Underflow	RO,LH	0x0	0x0	1 = Over/Underflow Error 0 = No FIFO Error
6:5	Reserved	RO	Always 00	Always 00	Reserved
4	Fiber Energy Detect Changed	RO,LH	0x0	0x0	1 = Energy Detect state changed 0 = No Energy Detect state change detected
3:0	Reserved	RO	Always 00000	Always 00000	Reserved

Table 405: Fiber Receive Error Counter Register
Page 1, Register 21

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Receive Error Count	RO, LH	0x0000	Retain	Counter will peg at 0xFFFF and will not roll over. Both False carrier and symbol errors are reported.

Table 406: Fiber Page Register
Page 1, Register 22

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	Page Select	R/W	0x0001	Retain	Page number for SERDES registers.



Table 407: PRBS Control
Page 1, Register 23

Bits	Field	Mode	HW Rst	SW Rst	Description
15:8	Reserved	R/W	0x00	Retain	Reserved
7	Invert Checker Polarity	R/W	0x0	Retain	0 = Invert 1 = Normal
6	Invert Generator Polarity	R/W	0x0	Retain	0 = Invert 1 = Normal
5	PRBS Lock	R/W	0x0	Retain	0 = Counter Free Runs 1 = Do not start counting until PRBS locks first
4	Clear Counter	R/W, SC	0x0	0x0	0 = Normal 1 = Clear Counter
3:2	Reserved	R/W	0x0	Retain	Reserved
1	PRBS Checker Enable	R/W	0x0	0x0	0 = Disable 1 = Enable
0	PRBS Generator Enable	R/W	0x0	0x0	0 = Disable 1 = Enable

Table 408: PRBS Error Counter LSB
Page 1, Register 24

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	PRBS Error Count LSB	RO	0x0000	Retain	A read to this register freezes register 25_1. Cleared only when register 23_1.4 is set to 1.

Table 409: PRBS Error Counter MSB
Page 1, Register 25

Bits	Field	Mode	HW Rst	SW Rst	Description
15:0	PRBS Error Count MSB	RO	0x0000	Retain	This register does not update unless register 24_1 is read first. Cleared only when register 23_1.4 is set to 1.

Table 410: Fiber Specific Control Register 2
Page 1, Register 26

Bits	Field	Mode	HW Rst	SW Rst	Description
15	Reference Clock Source Select	R/W	0x0	Retain	Changes to this bit are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect. 1 = Use SE_SCLK as 25 MHz source 0 = Use XTAL1 as 25 MHz source Changes to this bit do not take effect until a software reset is issued by setting Register 0_1.15 to 1.
14	1000BASE-X Noise Filtering	R/W	0x0	Retain	1 = Enable 0 = Disable
13	100BASE-FX Noise Filtering	R/W	0x0	Retain	1 = Enable 0 = Disable
12:10	Reserved	R/W	0x0	Update	Reserved
9	FEFI Enable	R/W	0x0	Retain	100BASE-FX FEFI 1 = Enable 0 = Disable
8	Reserved	R/W	0x0	Retain	Reserved
7	Reserved	R/W	0x0	Update	Reserved
6	Serial Interface Auto-Negotiation bypass enable	R/W	0x1	Update	Changes to this bit are disruptive to the normal operation; hence, any changes to these registers must be followed by software reset to take effect. 1 = Bypass Allowed 0 = No Bypass Allowed
5	Serial Interface Auto-Negotiation bypass status	RO	0x0	0x0	1 = Serial interface link came up because bypass mode timer timed out and fiber Auto-Negotiation was bypassed. 0 = Serial interface link came up because regular fiber Auto-Negotiation completed. If this bit is 1, then bit 17_1.11 will be 0.
4	Reserved	R/W	0x0	Update	Reserved
3	Fiber Transmitter Disable	R/W	0x0	Retain	1 = Transmitter Disable 0 = Transmitter Enable
2:0	SGMII/Fiber Output Amplitude	R/W	0x2	Retain	Differential voltage peak measured. See AC/DC section for valid VOD values. 000 = 14mV 001 = 112mV 010 = 210 mV 011 = 308mV 100 = 406mV 101 = 504mV 110 = 602mV 111 = 700mV



Link Street®

88E6321/88E6320 Functional Specification

12

EEPROM Programming Format

The device supports an optional external serial EEPROM device for programming its internal registers. The EEPROM data will be read in once after Reset is deasserted.

The device supports 1K bit (24C01), 2K bit (24C02) or 4K bit (24C04) 2-wire EEPROM devices. 2-Wire external EEPROM devices are treated as if they are in a x16 data organization by always reading both an odd address (as the upper 8-bits) and an even address (as the lower 8-bits).

No matter what device is attached, the EEPROM device is read and processed in the same way:

1. Start at EEPROM address 0x00.
2. Read in the 16-bits of data from the current address, this is called the Command.
3. If the just read in Command is all one's, terminate the serial EEPROM reading process, go to 8.
4. Increment the address by 1 (to the next address). If the Command does not need any data from the EEPROM, process the Command and go to step 2.
5. Read in the 16-bits of data from the next address, this is called RegData and increment the address by 1.
6. Write RegData to the location or locations defined by the previous Command.
7. Go to 2.
8. Set the EEInt bit in Global Status to a one (global 0x00) generating an Interrupt (if enabled).
9. Done.

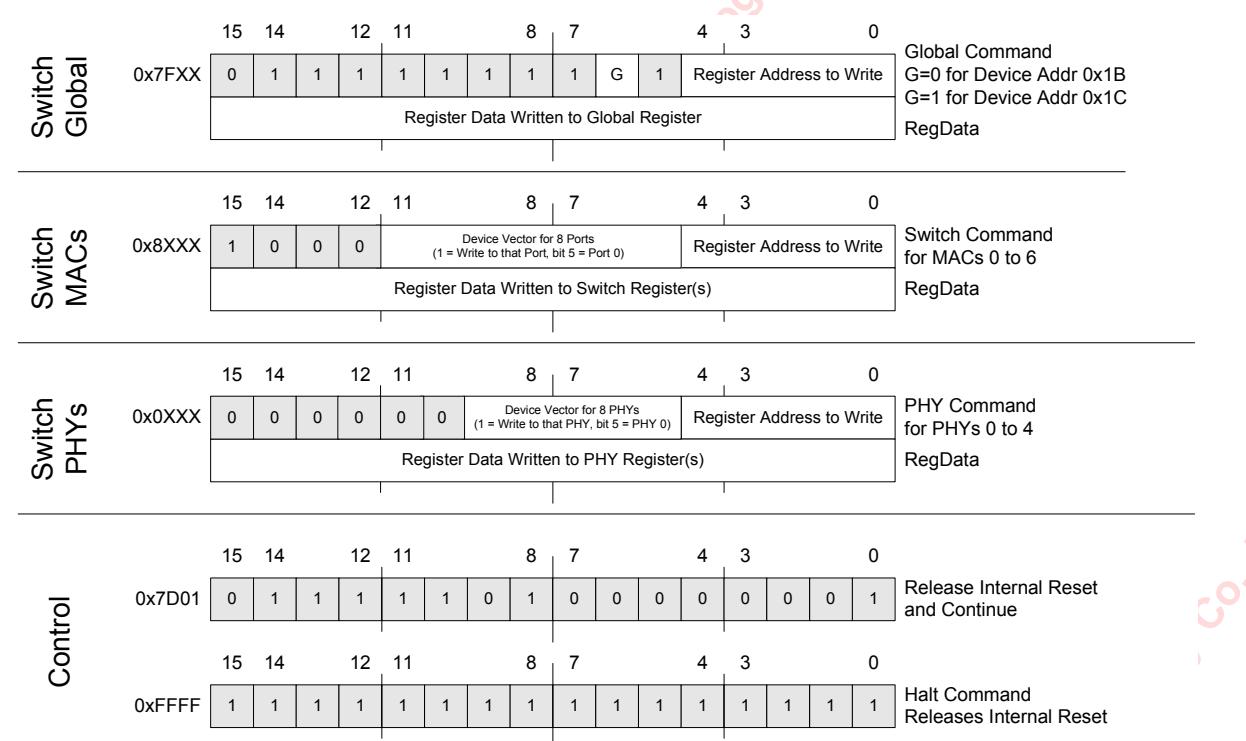
The 16-bit Command determines which register or registers inside the devices are updated as follows (refer to [Figure 77](#)):

1. Bit 15 determines which set of registers can be written. If bit 15 = 0 the five external PHY device's registers can be written to or the two Global register spaces can be written to (SMI Device Addresses 0x0F & 0x0E). If bit 15 = 1 the Switch registers can be written to (SMI Device Addresses 0x08 to 0x0D). See [Section 10](#) and [Figure 62](#) for more information on the registers and their addresses.
2. Bits 10:5 (or 9:5), the Device Vector, determines which Device Address or Addresses are written. Each bit of the Device Vector that is set to a one causes one Device Address to be written. Bit 5 controls writes for Port 0 (either PHY address 0x00 or Switch address 0x08). Bit 6 controls writes for Port 1 (either PHY address 0x01 or Switch address 0x09). Bits 7 controls writes for Switch Port 2. The Switch Global register set that is written to is determined by bit 6 the G bit. When G=1 the Global space at device address 0x0F is written to. When G=0 the Global 2 space at device address 0x0E is written to. Care is needed to insure Reserved registers are not written.
3. Bits 4:0, the Register Address, determine which SMI Register Address is written.
4. Two special commands are supported. Halt (end of EEPROM processing) and Release Internal Reset. Release Internal Reset is used to allow packets to start flowing (if Ports are in the Forwarding PortState) and to allow the EEPROM to be able to load the ATU and/or VTU as these blocks need internal reset released before they can be written to.

The format of the EEPROM Commands support writing the same RegData to all the PHY's or all the Switch's MAC's with one Command/RegData pair. The Command/RegData list can be as short or as long as needed. This makes optimum use of the limited number of Command/RegData pairs that can fit in a given size EEPROM (31¹ Command/RegData pairs in the 24C01, 63 in the 24C02 and 127 in the 24C04).

1. The maximum number of Command/RegData pairs is one less than expected because the last entry must be the End of List Indicator of all one's.

Figure 77: EEPROM Data Format





Marvell Semiconductor, Inc.
5488 Marvell Lane
Santa Clara, CA 95054, USA

Tel: 1.408.222.2500
Fax: 1.408.988.8279

www.marvell.com