



Programmer's Reference Guide

NetXtreme II

Standard Broadcom® NetXtreme® II Family Highly Integrated Media Access Controller

REVISION HISTORY

Revision	Date	Change Description
NetXtremell-PG203-R	10/08/08	<p>Updated:</p> <ul style="list-style-type: none">• "Introduction" on page 1• "Revision Levels" on page 8• Table 2, "Broadcom NetXtreme II Revision Levels," on page 8• Table 7, "Manufacturing Information Region," on page 38 and Table 10, "Licensing Information," on page 49• Table 49, "Programming the Host Coalescing Registers," on page 122• "CTX Command Register (ctx_command, Offset 0x1000)" on page 292 through "CTX Debug State Machine Register (ctx_debug_sm, offset 0x10a8)" on page 310• Updated and moved "Debug Port Access" on page 467• "MQ Debug Vector Peek Register (mq_debug_vect_peek, offset 0x3c94)" on page 417 through "P2R Command Register (p2r_command, Offset 0x244000)" on page 587 <p>Removed:</p> <ul style="list-style-type: none">• "Pseudo Code" on page 9 through "Transmit Data Path" on page 32• Table 54, "PCI Configuration Block (pci_config) Register Summary," on page 150• "Preamble" on page 1100 through "Data" on page 1102 <p>Added:</p> <ul style="list-style-type: none">• "BCM5716 Feature Description" on page 6• "Feature Comparison" on page 7• Figure 1, "Broadcom BCM5706 Block Diagram," on page 15 through Figure 3, "Broadcom BCM5709 and BCM5716 Block Diagram," on page 17• "Network Controller-Sideband Interface" on page 148• "Register Dependency Definitions" on page 149 through "Variable Register Definitions" on page 151• "RPM IPv6 Programmable Extension 0 Register (rpm_ipv6_programmable_extension0, offset 0x1854)" on page 348 through "RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl. Offset 0x2bfc)" on page 395

Revision	Date	Change Description
NetXtremell-PG202-R	03/14/07	<p>Updated:</p> <ul style="list-style-type: none">• Table 1: “Broadcom NetXtreme II Revision Levels,” on page 8 through Table 10: “Feature Configuration Information Region,” on page 46.• Figure 16: “NVRAM Memory Map,” on page 40.• “Host Address Field (rx_bd_haddr)” on page 89.• Section 9: “Driver/Firmware Shared Memory” on page 275 through “Driver/Firmware Mailboxes” on page 276.• Table 279: “Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68),” on page 311 and Table 280: “Miscellaneous Status Register (pcicfg_misc_status, Offset 0x6c),” on page 312.• Table 282: “Interrupt Acknowledge Command Register (pcicfg_int_ack_cmd, Offset 0x84),” on page 315.• Table 283: “MSI-X Control Register (pcicfg_msix_control, Offset 0xa2),” on page 317 through Table 309: “PCI Express Port Virtual Channel Resource Control Register (pcicfg_vc_rsrc_control, Offset 0x174),” on page 337.• Table 363: “MISC Command Register (misc_command, Offset 0x800),” on page 373 through Table 410: “cs16 err (0x8e0—BCM5709 Only),” on page 426.• Table 433: “perr status2 (0x94c—BCM5709 Only),” on page 441.• “Power Budgeting Enhanced Capability Header Register (epb_power_capability, Offset 0x14c)” on page 945 through “PIB Header Log Register (epb_pib_header_log, Offset 0x13c)” on page 944. <p>Removed:</p> <ul style="list-style-type: none">• “BOOTCODE State” on page 279. <p>Added:</p> <ul style="list-style-type: none">• “BCM5708C Feature Description” on page 3 through “BCM5708S Feature Description” on page 4.

Revision	Date	Change Description
NetXtremell-PG201-R	08/09/05	<p>Updated:</p> <ul style="list-style-type: none"> • Table 1: “Broadcom NetXtreme II Revision Levels,” on page 3. • Table 3: “Register Access Definitions,” on page 5. • Table 6: “Code Directory Region,” on page 35. • Table 7: “Manufacturing Information Region,” on page 37. • Table 8: “Feature Configuration Information Region,” on page 39. • Table 9: “Vital Product Data Information,” on page 43. • Table 10: “Licensing Information,” on page 44. • “NVRAM Pseudo Code” on page 48, entire section. • Table 18: “Flags Field (tx_bd_flags),” on page 82. • Headings in “On-Chip Memory Data Structures” on page 151. • Replaced Section 5: “Device Control” on page 248. • Table 275: “Interrupt Acknowledge Command Register (pcicfg_int_ack_cmd, Offset 0x84),” on page 305. • Table 277: “PCI GRC Window Address Register (grc_window_addr, Offset 0x400),” on page 308. • Table 328: “General Purpose Hardware Control 0 Register (gp_hw_ctl0, Offset 0x8bc),” on page 358. • Table 438: “RV2P Command Register (rv2p_command, Offset 0x2800),” on page 466. • Table 561: “NVM Command Register (nvm_command, Offset 0x6400),” on page 556. • “HC Attention Bits Enable Register (hc_attn_bits_enable, Offset 0x680c)” on page 567. • Transceiver register tables: Table 862: “MII Control Register (mii_ctrl, PHY_Addr = 0x1, Reg_Addr = 0x00),” on page 828 through Table 877: “MII Extended Status Register (mii_extend_stat, PHY_Addr = 0x1, Reg_Addr = 0x0F),” on page 842. • “PCI Configuration Block Registers” on page 285. <p>Added:</p> <ul style="list-style-type: none"> • “Physical Addressing Versus Logical Addressing” on page 46. • “L2 Context (l2_context)” on page 154. • Section 9: “Driver/Firmware Shared Memory” on page 270. • Section 10: “Firmware Features” on page 283. • “Receive DMA Registers” on page 481. • Section 12: “EPB Register Definitions” on page 798. <p>Removed:</p> <ul style="list-style-type: none"> • “Overview of Flash Interface” • “Universal Management Port Command Interface”
NetXtremell-PG200-R	03/08/05	Initial release.

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

© 2008 by Broadcom Corporation
All rights reserved
Printed in the U.S.A.

Broadcom®, the pulse logo, Connecting everything®, and the Connecting everything logo are among the trademarks of Broadcom Corporation and/or its affiliates in the United States, certain other countries and/or the EU. Any other trademarks or trade names mentioned are the property of their respective owners.

TABLE OF CONTENTS

Section 1: Introduction	1
BCM5706 Feature Description	1
BCM5708C Feature Description	2
BCM5708S Feature Description	3
BCM5709C Feature Description	4
BCM5709S Feature Description	5
BCM5716 Feature Description	6
Feature Comparison	7
Revision Levels	8
Notational Conventions	9
Sizes	9
Acronyms, Abbreviations, and Terms	10
Related Documentation	12
 Section 2: Hardware Architecture	14
Theory of Operations	14
Overview	18
TCP-Offload Engine	20
Remote DMA	22
<i>iSCSI</i>	24
Host Memory Data Structures	27
Buffer Descriptor Chain.....	28
Normal Record Entry	31
Multi-Record Entry	31
Next Pointer Record Entry	31
Page Tables	31
Page Table Format	31
Address Space	32
Host Memory Map.....	32
Internal Processor Address Space	33
 Section 3: NVRAM Configuration	35
NVRAM Map	35

Bootstrap	36
Code Directory.....	36
Manufacturing Information.....	38
Feature Configuration Information.....	42
Vital Product Data.....	48
Licensing Information	48
Program Images	51
Calculating the CRC-32 Checksum.....	51
Programming the Non-Volatile Memory	52
Physical Addressing Versus Logical Addressing	52
NVRAM Pseudo Code.....	53
Section 4: Common Data Structures.....	73
Host Memory Data Structures	73
Host Status Block (Status).....	73
Per Processor Status Block (status_per_cpu_block)	80
Host Statistics Block (Stat)	81
Receive Buffer Descriptor Chain Record (rx_bd)	90
Transmit Buffer Descriptor Chain Record (tx_bd)	91
RX Packet L2 Frame Status Header (l2_fhdr).....	92
On-Chip Memory Data Structures	95
L2 BD Chain Context (l2_bd_chain_context)	95
Neighbor State Context	96
L2 Context (l2_context)	98
RX Parser ACPI-WOL Pattern Format	112
Section 5: Device Control.....	114
Device Initialization Code Conventions.....	114
MAC Reset Procedure	116
MAC Initialization Procedure	118
CPU Initialization	124
RV2P Firmware Load Procedure and Pseudo Code.....	124
Shared Hardware Resources.....	126
Section 6: PCI	127
Expansion ROM	127

Description	127
Operational Characteristics.....	127
BIOS	127
Byte-Swapping Issues	128
Background.....	128
Architecture	129
Byte-Swapping Versus Word-Swapping	130
Selecting the Correct Swapping Modes	131
Enabling Swapping for PCI Target Accesses	131
Enabling Swapping for DMA Accesses.....	132
Section 7: Interrupt Processing.....	133
Interrupt Processing	133
Host Coalescing	133
Message Signaled Interrupts	134
Attentions	134
Interrupt Procedure	134
Section 8: Driver/Firmware Shared Memory	135
Driver/Firmware Mailboxes	136
Driver Reset Signature Mailbox	137
Driver Mailbox (drv_mb).....	137
Driver Pulse Mailbox (drv_pulse_mb)	138
Firmware Mailbox (fw_mb).....	138
Firmware Synchronization Pseudo Code.....	138
Device Information (dev_info).....	139
Shared Hardware Configuration (shared_hw_cfg).....	141
Port-Specific Hardware Configuration (port_hw_cfg).....	142
Shared Feature Configuration (shared_feat_cfg)	142
Port Feature Configuration (port_feat_cfg)	143
Software License Key (license_key)	145
Management Firmware State (mgmtfw_state).....	146
Section 9: Firmware Features.....	147
Bootcode.....	147
Intelligent Platform Management Initiative	148

Universal Management Port	148
Network Controller-Sideband Interface	148
Section 10: MAC Register Definitions	149
Register Access Definitions	149
Register Dependency Definitions	149
Register Reset Values	150
Variable Register Definitions	151
PCI Configuration Block Registers	152
Vendor ID Register (pcicfg_vendor_id, Offset 0x00)	152
Device ID Register (pcicfg_device_id, Offset 0x02)	152
Command Register (pcicfg_command, Offset 0x04).....	153
Status Register (pcicfg_status, Offset 0x06)	154
Revision ID Register (pcicfg_revision_id, Offset 0x08)	155
Class Code Register (pcicfg_class_code, Offset 0x09)	155
Cache Line Size Register (pcicfg_cache_line_size, Offset 0x0c)	155
Latency Timer Register (pcicfg_latency_timer, Offset 0xd).....	156
Header Type Register (pcicfg_header_type, Offset 0xe)	156
BIST Register (pcicfg_bist, Offset 0xf)	156
Base Address 1 Register (pcicfg_bar_1, Offset 0x10)	156
Base Address 2 Register (pcicfg_bar_2, Offset 0x14)	157
Base Address 3Register (pcicfg_bar_3, Offset 0x18)	157
Base Address 4 Register (pcicfg_bar_4, Offset 0x1c).....	158
Subsystem Vendor ID Register (pcicfg_subsystem_vendor_id, Offset 0x2c)	159
Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e).....	159
Expansion ROM Base Address Register (pcicfg_exp_rom_bar, Offset 0x30)	159
Capabilities Pointer Register (pcicfg_cap_pointer, Offset 0x34)	160
Interrupt Line Register (pcicfg_int_line, Offset 0x3c).....	160
Interrupt Pin Register (pcicfg_int_pin, Offset 0x3d).....	160
Minimum Grant Register (pcicfg_min_grant, Offset 0x3e)	161
Maximum Latency Register (pcicfg_maximum_latency, Offset 0x3f).....	161
PCI-X Capabilities Registers	161
PCI-X Capabilities Register (pcicfg_pcix_cap_id, Offset 0x40)	161
PCI-X Next Capability Pointer Register (pcicfg_pcix_next_cap_ptr, Offset 0x41).....	161
PCI-X Command Register (pcicfg_pcix_command, Offset 0x42).....	162

PCI-X Status Register (pcicfg_pcix_status, Offset 0x44).....	163
PCI Power Management Registers	165
Power Management Capability ID Register (pcicfg_pm_cap_id, Offset 0x48)	165
Power Management Next Capability Register (pcicfg_pm_next_cap_ptr, 0x49)	165
Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a).....	166
Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c).....	167
Power Management C/S BSE Register (pcicfg_pm_csr_bse, Offset 0x4e).....	168
Power Management Data Register (pcicfg_pm_data, Offset 0x4f).....	168
Vital Product Data Capabilities Registers	169
VPD Capability ID Register (pcicfg_vpd_cap_id, Offset 0x50)	169
VPD Next Capability Pointer Register (pcicfg_vpd_next_cap_ptr, Offset 0x51)	169
VPD Address/Flag Register (pcicfg_vpd_flag_addr, Offset 0x52)	169
VPD Data Register (pcicfg_vpd_data, Offset 0x54).....	169
Message Signaled Interrupts Registers	170
MSI Capability ID Register (pcicfg_msi_cap_id, 0x58)	170
MSI Next Capability Pointer Register (pcicfg_msi_next_cap_ptr, Offset 0x59)	170
MSI Control Register (pcicfg_msi_control, Offset 0x5a)	170
MSI Address Low Register (pcicfg_msi_addr_l, Offset 0x5c).....	171
MSI Address High Register (pcicfg_msi_addr_h, Offset 0x60)	171
MSI Data Register (pcicfg_msi_data, Offset 0x64).....	171
Private Configuration Registers	172
Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68)	172
Miscellaneous Status Register (pcicfg_misc_status, Offset 0x6c).....	173
PCI Clock Control Register (pcicfg_pci_clock_control_bits, Offset 0x70).....	174
Register Window Address Register (pcicfg_reg_window_address, Offset 0x78)	176
Register Window Data Register (pcicfg_reg_window, Offset 0x80).....	176
Interrupt Acknowledge Command Register (pcicfg_int_ack_cmd, Offset 0x84)	177
Status Bit Set Command Register (pcicfg_status_bit_set_cmd, Offset 0x88)	178
Status Bit Clear Command Register (pcicfg_status_bit_clear_cmd, Offset 0x8c)	178
Mailbox Queue Address Register (pcicfg_mailbox_queue_addr, Offset 0x90).....	178
Mailbox Queue Data Register (pcicfg_mailbox_queue_data, Offset 0x94)	178
MSI-X Capabilities Registers	179
MSI-X Capability ID Register (pcicfg_msix_cap_id, Offset 0xa0)	179
MSI-X Next Capability Pointer Register (pcicfg_msix_next_cap_ptr, Offset 0xa1)	179
MSI-X Control Register (pcicfg_msix_control, Offset 0xa2)	179

MSI-X Table Offset Base Index Register (pcicfg_msix_tbl_off Bir, Offset 0xa4)	180
MSI-X PBA Offset Base Register (pcicfg_msix_pba_off Bir, Offset 0xa8)	180
PCI Express Capabilities Registers	181
PCI Express Capability ID Register (pcicfg_PCIE_cap_id, Offset 0xac)	181
PCI Express Next Capability Pointer Register (pcicfg_PCIE_next_cap_ptr, Offset 0xad)	181
PCI Express Capabilities Register (pcicfg_PCIE_capability, Offset 0xae)	181
PCI Express Device Capabilities Register (pcicfg_device_capability, Offset 0xb0)	181
PCI Express Device Control Register (pcicfg_device_control, Offset 0xb4)	182
PCI Express Device Status Register (pcicfg_device_status, Offset 0xb6)	183
PCI Express Link Capability Register (pcicfg_link_capability, Offset 0xb8)	184
PCI Express Link Control Register (pcicfg_link_control, Offset 0xbc)	185
PCI Express Link Status Register (pcicfg_link_status, Offset 0xbe)	185
PCI Express Slot Capability Register (pcicfg_slot_capability, Offset 0xc0)	186
PCI Express Slot Control Register (pcicfg_slot_control, Offset 0xc4)	186
PCI Express Slot Status Register (pcicfg_slot_status, Offset 0xc6)	186
PCI Express Root Control Register (pcicfg_root_control, Offset 0xc8)	186
PCI Express Root Capabilities Register (pcicfg_root_cap, Offset 0xca)	186
PCI Express Root Status Register (pcicfg_root_status, Offset 0xcc)	186
PCI Express Device Capability 2 Register (pcicfg_device_capability_2, Offset 0xd0)	187
PCI Express Device Control 2 Register (pcicfg_device_control_2, Offset 0xd4)	187
PCI Express Device Status 2 Register (pcicfg_device_status_2, Offset 0xd6)	188
PCI Express Link Capability 2 Register (pcicfg_link_capability_2, Offset 0xd8)	188
PCI Express Link Control 2 Register (pcicfg_link_control_2, Offset 0xdc)	188
PCI Express Link Status 2 Register (pcicfg_link_status_2, Offset 0xde)	189
PCI Express Device Serial Number Capabilities Registers	191
PCI Express Device Serial Number Capability ID Register (pcicfg_device_ser_num_cap_id, Offset 0x100)	191
PCI Express Device Serial Number Next Capability Pointer Register (pcicfg_device_ser_num_cap_off, Offset 0x102)	191
PCI Express Lower Serial Number Register (pcicfg_lower_ser_num, Offset 0x104)	191
PCI Express Upper Serial Number Register (pcicfg_upper_ser_num, Offset 0x108)	191
PCI Express Advanced Error Capabilities Registers	192
PCI Express Advanced Error Capability ID Register (pcicfg_adv_err_cap_id, Offset 0x110)	192
PCI Express Advanced Error Next Capability Pointer Register (pcicfg_adv_err_cap_off, Offset 0x112)	192
PCI Express Uncorrectable Error Status Register (pcicfg_ucorr_err_status, Offset 0x114)	192

PCI Express Uncorrectable Error Mask Register (pcicfg_uecc_err_mask, Offset 0x118)	193
PCI Express Uncorrectable Error Severity Register (pcicfg_uecc_err_sevr, Offset 0x11c)	193
PCI Express Correctable Error Status Register (pcicfg_cecc_err_status, Offset 0x120).....	194
PCI Express Correctable Error Mask Register (pcicfg_cecc_err_mask, Offset 0x124)	194
PCI Express Advanced Error Capability Control Register (pcicfg_aecc_cap_control, Offset 0x128) ..	195
PCI Express Header Log 1 Register (pcicfg_header_log1, Offset 0x12c).....	195
PCI Express Header Log 2 Register (pcicfg_header_log2, Offset 0x130).....	195
PCI Express Header Log 3 Register (pcicfg_header_log3, Offset 0x134).....	195
PCI Express Header Log 4 Register (pcicfg_header_log4, Offset 0x138).....	195
PCI Express Power Budget Capability ID Register (pcicfg_pwr_bdgt_cap_id, Offset 0x150	195
PCI Express Power Budget Capabilities Registers	196
PCI Express Power Budget Next Capability Pointer Register (pcicfg_pwr_bdgt_cap_off, Offset 0x152).	196
PCI Express Power Budget Data Select Register (pcicfg_pwr_bdgt_data_sel, Offset 0x154).....	196
PCI Express Power Budget Data Register (pcicfg_pwr_bdgt_data, Offset 0x158).....	197
PCI Express Power Budget Capability Register (pcicfg_pwr_bdgt_capability, Offset 0x15c).....	197
PCI Express Virtual Channel Capabilities Registers	198
PCI Express Virtual Channel Capability ID Register (vc_cap_id, Offset 0x160).....	198
PCI Express Virtual Channel Next Capability Pointer Register (pcicfg_vc_cap_off, Offset 0x162)	198
PCI Express Port Virtual Channel Capabilities Registers	199
PCI Express Port Virtual Channel Capability Register (pcicfg_port_vc_capability, Offset 0x164).....	199
PCI Express Port Virtual Channel Capability 2 Register (pcicfg_port_vc_capability2, Offset 0x168). 199	199
PCI Express Port Virtual Channel Control Register (pcicfg_port_vc_control, Offset 0x16c)	199
PCI Express Port Virtual Channel Status Register (pcicfg_port_vc_status, Offset 0x16e).....	200
PCI Express Port Virtual Channel Arbitration Table Register (pcicfg_port_arb_table, Offset 0x170). 200	200
PCI Express Port Virtual Channel Resource Control Register (pcicfg_vc_rsrc_control, Offset 0x174)	200
PCI Interface Block Registers	201
PCI GRC Window Address Register (pci_grc_window_addr, Offset 0x400)	202
PCI Configuration 1 Register (pci_config_1, Offset 0x404)	202
PCI Configuration 2 Register (pci_config_2, Offset 0x408)	204
PCI Configuration 3 Register (pci_config_3, Offset 0x40c).....	207
PCI Power Management Data A Register (pci_pm_data_a, Offset 0x410)	208
PCI Power Management Data B Register (pci_pm_data_b, Offset 0x414)	209
PCI Swap Diagnostics 0 Register (pci_swap_diag0, Offset 0x418).....	209

PCI Swap Diagnostics 1 Register (pci_swap_diag1, Offset 0x41c)	209
PCI Expansion ROM Address Register (pci_exp_rom_addr, Offset 0x420)	210
PCI Expansion ROM Data Register (pci_exp_rom_data, Offset 0x424)	210
PCI VPD Interface Register (pci_vpd_intf, Offset 0x428)	210
PCI VPD Address Flag Register (pci_vpd_addr_flag, Offset 0x42e)	211
PCI VPD Data Register (pci_vpd_data, Offset 0x430)	211
PCI ID Value 1 Register (pci_id_val1, Offset 0x434)	211
PCI ID Value 2 Register (pci_id_val2, Offset 0x438)	212
PCI ID Value 3 Register (pci_id_val3, Offset 0x43c)	213
PCI ID Value 4 Register (pci_id_val4, Offset 0x440)	214
PCI ID Value 5 Register (pci_id_val5, Offset 0x444)	217
PCI PCI-X Extended Status Register (pci_pcix_extended_status, Offset 0x448)	218
PCI ID Value 6 Register (pci_id_val6, Offset 0x44c)	219
PCI MSI Data Register (pci_msi_data, Offset 0x450)	219
PCI MSI Address High Register (pci_msi_addr_h, Offset 0x454)	219
PCI MSI Address Lo Register (pci_msi_addr_l, Offset 0x458)	219
PCI Configuration Access Command Register (pci_cfg_access_cmd, Offset 0x45c)	220
Miscellaneous (MISC) Block Registers.....	221
MISC Command Register (misc_command, Offset 0x800)	221
MISC Configuration Register (misc_cfg, Offset 0x804)	224
MISC ID Register (misc_id, Offset 0x808)	229
MISC Enable Status Register (misc_enable_status_bits, Offset 0x80c)	230
MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)	232
MISC Enable Clear Register (misc_enable_clr_bits, Offset 0x814)	234
MISC Clock Control Register (misc_clock_control_bits, Offset 0x818)	236
MISC SPIO Register (misc_spio, Offset 0x81c)	240
MISC SPIO Interrupt Register (misc_spio_int, Offset 0x820)	241
MISC LFSR Configuration Register (misc_config_lfsr, Offset 0x824)	242
MISC LFSR Mask Register (misc_lfsr_mask_bits, Offset 0x828)	242
MISC Arbitration Request Registers (misc_arb_req[5], Offset 0x82c)	244
MISC Arbitration Free Registers (misc_arb_free[5], Offset 0x840)	245
MISC Arbitration Request Status Registers (misc_arb_req_status[5], Offset 0x854)	245
MISC Arbitration Grant 0 Register (misc_arb_gnt0, Offset 0x868)	246
MISC Arbitration Grant 1 Register (misc_arb_gnt1, Offset 0x86c)	247
MISC Arbitration Grant 2 Register (misc_arb_gnt2, 0x870)	248

MISC Arbitration Grant 3 Register (misc_arb_gnt3, Offset 0x874)	249
MISC PRBS Control Register (misc_prbs_control, Offset 0x878)	250
MISC PRBS Status Register (misc_prbs_status, Offset 0x87c)	250
MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)	251
MISC SMBus In Register (misc_smb_in, Offset 0x884)	253
MISC SMBus Out Register (misc_smb_out, Offset 0x888)	254
MISC SMBus Watchdog Register (misc_smb_watchdog, Offset 0x88c)	256
MISC SMBus Heartbeat Register (misc_smb_heartbeat, Offset 0x890)	256
MISC SMBus Poll ASF Register (misc_smb_poll_asf, Offset 0x894)	257
MISC SMBus Poll Legacy Register (misc_smb_poll_legacy, Offset 0x898)	257
MISC SMBus Retransmit Register (misc_smb_retran, Offset 0x89c)	257
MISC SMBus Timestamp Register (misc_smb_timestamp, Offset 0x8a0)	258
MISC PERR Enable 0 Register (misc_perr_ena0, Offset 0x8a4)	258
MISC PERR Enable 1 Register (misc_perr_ena1, Offset 0x8a8)	260
MISC PERR Enable 2 Register (misc_perr_ena2, Offset 0x8ac)	261
MISC Voltage Regulator Control Register (misc_vreg_control, Offset 0x8b4)	262
MISC Final Clock Control Value Register (misc_vreg_control, Offset 0x8b8)	264
General-Purpose Hardware Control 0 Register (gp_hw_ctl0, Offset 0x8bc)	264
General-Purpose Hardware Control 1 Register (gp_hw_ctl1, Offset 0x8c0)	268
MISC New Core Control Register (misc_new_core_ctl, offset 0x8c8)	269
MISC Checksum 16 Error Register (misc_cs16_err, offset 0x8e0)	270
MISC Dual Media Control Register (misc_dual_media_ctrl, offset 0x8ec)	271
MISC BIST CS2 Register (misc_bist_cs2, offset 0x91c)	274
MISC Parity Error Status Register (misc_perr_status0, offset 0x944)	275
MISC Parity Error Status 1 Register (misc_perr_status1, offset 0x948)	276
MISC Parity Error Status 2 Register (misc_perr_status2, offset 0x94c)	277
DMA Engine (DMA) Registers	278
DMA Command Register (dma_command, Offset 0xc00)	278
DMA Status Register (dma_status, Offset 0xc04)	279
DMA Configuration Register (dma_config, Offset 0xc08)	281
DMA Blackout Register (dma_blackout, Offset 0xc0c)	283
DMA Read Master Setting 0 Register (dma_read_master_setting_0, offset 0xc10)	285
DMA Read Master Setting 1 Register (dma_read_master_setting_1, 0xc14)	286
DMA Write Master Setting 0 Register (dma_write_master_setting_0, offset 0xc18)	287
DMA Write Master Setting 1 Register (dma_write_master_setting_1, offset 0xc1c)	288

DMA Arbiter Register (dma_arbiter, offset 0xc20).....	289
DMA Arbiter Timers Register (dma_arb_timers, offset 0xc24).....	291
Context Memory (CTX) Registers.....	292
CTX Command Register (ctx_command, Offset 0x1000)	292
CTX Status Register (ctx_status, Offset 0x1004)	294
CTX Virtual Address Register (ctx_virt_addr, Offset 0x1008)	297
CTX Page Table Register (ctx_page_tbl, Offset 0x100c).....	297
CTX Data Address Register (ctx_data_adr, Offset 0x1010).....	297
CTX Data Register (ctx_data, Offset 0x1014).....	298
CTX Lock Register (ctx_lock, Offset 0x1018).....	298
CTX Control Register (ctx_ctrl, offset 0x101c)	300
CTX CTX Data Register (ctx_ctx_data, offset 0x1020)	301
CTX Access Status Register (ctx_access_status, Offset 0x1040)	301
CTX Debug Lock Status Register (ctx_dbg_lock_status, Offset 0x1044)	302
CTX Cache Control Status Register (ctx_cache_ctrl_status, Offset 0x1048)	302
CTX Cache Control State Machine Status Register (ctx_cache_ctrl_sm_status, Offset 0x104c).....	303
CTX Cache Status Register (ctx_cache_status, Offset 0x1050).....	304
CTX DMA Status Register (ctx_dma_status, Offset 0x1054).....	304
CTX Report Status Register (ctx_rep_status, Offset 0x1058).....	306
CTX Checksum Error Status Register (ctx_cksum_error_status, Offset 0x105c)	308
CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080).....	308
CTX Channel Lock Status 1 Register (ctx_chnl_lock_status_1, Offset 0x1084).....	308
CTX Channel Lock Status 2 Register (ctx_chnl_lock_status_2, Offset 0x1088).....	308
CTX Channel Lock Status 3 Register (ctx_chnl_lock_status_3, Offset 0x108c)	309
CTX Channel Lock Status 4 Register (ctx_chnl_lock_status_4, Offset 0x1090).....	309
CTX Channel Lock Status 5 Register (ctx_chnl_lock_status_5, Offset 0x1094)	309
CTX Channel Lock Status 6 Register (ctx_chnl_lock_status_6, Offset 0x1098)	309
CTX Channel Lock Status 7 Register (ctx_chnl_lock_status_7, Offset 0x109c)	309
CTX Channel Lock Status 8 Register (ctx_chnl_lock_status_8, Offset 0x10a0)	309
CTX Channel Lock Status 9 Register (ctx_chnl_lock_status_9, Offset 0x10a4)	310
CTX Debug State Machine Register (ctx_debug_sm, offset 0x10a8)	310
CTX Cache Control Register (ctx_cache_ctrl, offset 0x10c0)	310
CTX Cache Data Register (ctx_cache_data, offset 0x10c4)	310
CTX Host Page Table Control Register (ctx_host_page_tbl_ctrl, offset 0x10c8).....	310
CTX Host Page Table Data 0 Register (ctx_host_page_tbl_data0, offset 0x10cc)	311

CTX Host Page Table Data 1 Register (ctx_host_page_tbl_data1, offset 0x10d0).....	311
Ethernet MAC (EMAC) Registers	312
EMAC Mode Register (emac_mode, Offset 0x1400).....	312
EMAC Status Register (emac_status, Offset 0x1404).....	314
EMAC Attention Enable Register (emac_attention_ena, Offset 0x1408).....	316
EMAC LED Register (emac_led, Offset 0x140c)	317
EMAC MAC Match Registers (emac_mac_match[32], Offset 0x1410).....	319
EMAC Backoff Seed Register (emac_backoff_seed, Offset 0x1498).....	319
EMAC RX MTU Size Register (emac_rx_mtu_size, Offset 0x149c).....	320
EMAC MDIO Command Register (emac_mdio_comm, Offset 0x14ac)	321
EMAC MDIO Status Register (emac_mdio_status, Offset 0x14b0).....	323
EMAC MDIO Mode Register (emac_mdio_mode, Offset 0x14b4).....	323
EMAC MDIO Auto Poll Status Register (emac_mdio_auto_status, Offset 0x14b8)	324
EMAC TX Mode Register (emac_tx_mode, Offset 0x14bc).....	325
EMAC TX Status Register (emac_tx_status, Offset 0x14c0).....	326
EMAC TX Lengths Register (emac_tx_lengths, Offset 0x14c4)	326
EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8).....	327
EMAC RX Status Register (emac_rx_status, Offset 0x14cc)	329
EMAC Multicast Hash Registers (emac_multicast_hash[8], Offset 0x14d0)	329
EMAC Checksum Error Status Register (emac_cksum_error_status, Offset 0x14f0)	329
EMAC ifHcInOctets Register (emac_rx_stat_ifhcinoctets, Offset 0x1500)	329
EMAC ifHcInBadOctets Register (emac_rx_stat_ifhcinbadoctets, Offset 0x1504)	330
EMAC etherStatsFragments Register (emac_rx_stat_etherstatsfragments, Offset 0x1508).....	330
EMAC ifHcInUnicastPkts Register (emac_rx_stat_ifhcinunicastpkts, Offset 0x150c)	330
EMAC ifHcInMulticastPkts Register (emac_rx_stat_ifhcinmulticastpkts, Offset 0x1510)	330
EMAC ifHcInBroadcastPkts Register (emac_rx_stat_ifhcinbroadcastpkts, Offset 0x1514).....	330
EMAC dot3StatsFCSErrors Register (emac_rx_stat_dot3statsfcserrors, Offset 0x1518)	331
EMAC dot3StatsAlignmentErrors Register (emac_rx_stat_dot3statsalignmenterrors, Offset 0x151c)....	331
EMAC dot3StatsCarrierSenseErrors Register (emac_rx_stat_dot3statscarriersenseerrors, Offset 0x1520)	331
EMAC xonPauseFramesReceived Register (emac_rx_stat_xonpauseframesreceived, Offset 0x1524)..	331
EMAC xoffPauseFramesReceived Register (emac_rx_stat_xoffpauseframesreceived, Offset 0x1528)..	331
EMAC macControlFramesReceived Register (emac_rx_stat_maccontrolframesreceived, Offset	

0x152c).....	332
EMAC xoffStateEntered Register (emac_rx_stat_xoffstateentered, Offset 0x1530).....	332
EMAC dot3StatsFramesTooLong Register (emac_rx_stat_dot3statsframestoolong, Offset 0x1534)	332
EMAC etherStatsJabbers Register (emac_rx_stat_etherstatsjabbers, Offset 0x1538).....	332
EMAC etherStatsUnderSizePkts Register (emac_rx_stat_etherstatsundersizepkts, Offset 0x153c)	332
EMAC etherStatsPkts64Octets Register (emac_rx_stat_etherstatspkts64octets, Offset 0x1540).....	333
EMAC etherStatsPkts65OctetsTo127Octets Register (emac_rx_stat_etherstatspkts65octetsto127octets, Offset 0x1544).....	333
EMAC etherStatsPkts128OctetsTo255Octets Register (emac_rx_stat_etherstatspkts128octetsto255octets, Offset 0x1548).....	333
EMAC etherStatsPkts256OctetsTo511Octets Register (emac_rx_stat_etherstatspkts256octetsto511octets, Offset 0x154c).....	333
EMAC etherStatsPkts512OctetsTo1023Octets Register (emac_rx_stat_etherstatspkts512octetsto1023octets, Offset 0x1550).....	333
EMAC etherStatsPkts1024OctetsTo1522Octets Register (emac_rx_stat_etherstatspkts1024octetsto1522octets, Offset 0x1554).....	333
EMAC etherStatsPkts1523OctetsTo9022Octets Register (emac_rx_stat_etherstatspkts1523octetsto9022octets, Offset 0x1558).....	334
EMAC Receive Statistics False Carrier Errors Register (emac_rx_stat_falsecarriererrors, Offset 0x1574) 334	
EMAC Receive Statistics Auto-Clear Registers (emac_rx_stat_ac[23], Offset 0x1580)	334
EMAC Receive Statistics Register (emac_rx_stat_ac_28, Offset 0x15f4)	334
EMAC ifHcOutOctets Register (emac_tx_stat_ifhcoutoctets, Offset 0x1600).....	334
EMAC ifHcOutBadOctets Register (emac_tx_stat_ifhcoutbadoctets, Offset 0x1604).....	334
EMAC etherStatsCollisions Register (emac_tx_stat_etherstatscollisions, Offset 0x1608)	335
EMAC outXonSent Register (emac_tx_stat_outxonsent, Offset 0x160c)	335
EMAC outXoffSent Register (emac_tx_stat_outxoffsent, Offset 0x1610)	335
EMAC flowControlDone Register (emac_tx_stat_flowcontroldone, Offset 0x1614).....	335
EMAC dot3StatsSingleCollisionFrames Register (emac_tx_stat_dot3statssinglecollisionframes, Offset 0x1618).....	335
EMAC dot3StatsMultipleCollisionFrames Register (emac_tx_stat_dot3statsmultiplecollisionframes, Offset 0x161c)	336
EMAC dot3StatsDeferredTransmissions Register (emac_tx_stat_dot3statsdeferredtransmissions, Offset 0x1620)	336
EMAC dot3StatsExcessiveCollisions Register (emac_tx_stat_dot3statsexcessivecollisions, Offset 0x1624).....	337
EMAC dot3StatsLateCollisions Register (emac_tx_stat_dot3statslatecollisions, Offset 0x1628)	337
EMAC ifHcOutUnicastPkts Register (emac_tx_stat_ifhcoutunicastpkts, Offset 0x162c)	337

EMAC ifHcOutMulticastPkts Register (emac_tx_stat_ifhcoutmulticastpkts, Offset 0x1630).....	337
EMAC ifHcOutBroadcastPkts Register (emac_tx_stat_ifhcoutbroadcastpkts, Offset 0x1634).....	337
EMAC etherStatsPkts64Octets Register (emac_tx_stat_etherstatspkts64octets, Offset 0x1638)	337
EMAC etherStatsPkts65OctetsTo127Octets Register (emac_tx_stat_etherstatspkts65octetsto127octets, Offset 0x163c).....	338
EMAC etherStatsPkts128OctetsTo255Octets Register (emac_tx_stat_etherstatspkts128octetsto255octets, Offset 0x1640)	338
EMAC etherStatsPkts256OctetsTo511Octets Register (emac_tx_stat_etherstatspkts256octetsto511octets, Offset 0x1644)	338
EMAC etherStatsPkts512OctetsTo1023Octets Register (emac_tx_stat_etherstatspkts512octetsto1023octets, Offset 0x1648)	338
EMAC etherStatsPkts1024OctetsTo1522Octets Register (emac_tx_stat_etherstatspkts1024octetsto1522octets, Offset 0x164c).....	338
EMAC etherStatsPkts1522OctetsTo9022Octets Register (emac_tx_stat_etherstatspkts1523octetsto9022octets, Offset 0x1650)	338
EMAC dot3StatsInternalMacTransmitErrors Register (emac_tx_stat_dot3statsinternalmactransmiterrors, Offset 0x1654).....	339
EMAC Transmit Statistics Auto-Clear Registers (emac_tx_stat_ac[22], Offset 0x1680)	339
EMAC Transmit Rate Limit Control Register (emac_tx_rate_limit_ctrl, Offset 0x16fc)	340
Receive Parser MAC (RPM) Registers.....	341
RPM Command Register (rpm_command, Offset 0x1800)	341
RPM Status Register (rpm_status, Offset 0x1804)	341
RPM Configuration Register (rpm_config, Offset 0x1808).....	342
RPM Management Packet Control Register (rpm_mgmt_pkt_ctrl, offset 0x180c).....	343
RPM VLAN Match 0 Register (rpm_vlan_match0, Offset 0x1810)	343
RPM VLAN Match 1 Register (rpm_vlan_match1, Offset 0x1814)	344
RPM VLAN Match 2 Register (rpm_vlan_match2, Offset 0x1818)	344
RPM VLAN Match 3 Register (rpm_vlan_match3, Offset 0x181c)	344
RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820).....	345
RPM Sort User 1 Register (rpm_sort_user1, Offset 0x1824).....	346
RPM Sort User 2 Register (rpm_sort_user2, Offset 0x1828).....	346
RPM Sort User 3 Register (rpm_sort_user3, Offset 0x182c).....	346
RPM L2 Filter Discards Statistic Register (rpm_stat_l2_filter_discards, Offset 0x1840)	347
RPM Rule Check Discards Statistic Register (rpm_stat_rule_checker_discards, Offset 0x1844).....	347
RPM ifInFtqDiscards Statistic Register (rpm_stat_ifinftqdiscards, Offset 0x1848)	347
RPM ifInMbufDiscard Statistic Register (rpm_stat_ifinmbufdiscard, Offset 0x184c)	347
RPM Rule Checker P4 Hits Statistic Register (rpm_stat_rule_checker_p4_hit, Offset 0x1850).....	347

RPM IPv6 Programmable Extension 0 Register (rpm_ipv6_programmable_extension0, offset 0x1854) .	348
RPM IPv6 Programmable Extension 1 Register (rpm_ipv6_programmable_extension1, offset 0x1858) .	348
RPM IPv6 Programmable Extension 2 Register (rpm_ipv6_programmable_extension2, offset 0x185c) .	348
RPM IPv6 Programmable Extension 3 Register (rpm_ipv6_programmable_extension3, offset 0x1860) .	348
RPM IPv6 Programmable Extension 4 Register (rpm_ipv6_programmable_extension4, offset 0x1864) .	349
RPM IPv6 Programmable Extension 5 Register (rpm_ipv6_programmable_extension5, offset 0x1868) .	349
RPM IPv6 Programmable Extension 6 Register (rpm_ipv6_programmable_extension6, offset 0x186c) .	349
RPM IPv6 Programmable Extension 7 Register (rpm_ipv6_programmable_extension7, offset 0x1870) .	349
RPM Statistics Auto-Clear Registers (rpm_stat_ac[5], Offset 0x1880)	349
RPM Rule Check Control 16 Register (rpm_rc_cntl_16, offset 0x18e0)	349
RPM Rule Check Value/Mask 16 Register (rpm_rc_value_mask_16, offset 0x18e4)	350
RPM Rule Check Control 17 Register (rpm_rc_cntl_17, offset 0x18e8)	350
RPM Rule Check Value/Mask 17 Register (rpm_rc_value_mask_17, offset 0x18ec).....	350
RPM Rule Check Control 18 Register (rpm_rc_cntl_18, offset 0x18f0)	350
RPM Rule Check Value/Mask 18 Register (rpm_rc_value_mask_18, offset 0x18f4)	350
RPM Rule Check Control 19 Register (rpm_rc_cntl_19, offset 0x18f8)	350
RPM Rule Check Value/Mask 19 Register (rpm_rc_value_mask_19, offset 0x18fc).....	351
RPM Rule Check Control 0 Register (rpm_rc_cntl_0, Offset 0x1900)	351
RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904).....	354
RPM Rule Check Control 1 Register (rpm_rc_cntl_1, Offset 0x1908)	354
RPM Rule Check Value/Mask 1 Register (rpm_rc_value_mask_1, Offset 0x190c).....	354
RPM Rule Check Control 2 Register (rpm_rc_cntl_2, Offset 0x1910)	354
RPM Rule Check Value/Mask 2 Register (rpm_rc_value_mask_2, Offset 0x1914).....	354
RPM Rule Check Control 3 Register (rpm_rc_cntl_3, Offset 0x1918)	355
RPM Rule Check Value/Mask 3 Register (rpm_rc_value_mask_3, Offset 0x191c).....	355
RPM Rule Check Control 4 Register (rpm_rc_cntl_4, Offset 0x1920)	355
RPM Rule Check Value/Mask 4 Register (rpm_rc_value_mask_4, Offset 0x1924).....	355
RPM Rule Check Control 5 Register (rpm_rc_cntl_5, Offset 0x1928)	355
RPM Rule Check Value/Mask 5 Register (rpm_rc_value_mask_5, Offset 0x192c).....	355

RPM Rule Check Control 6 Register (rpm_rc_cntl_6, Offset 0x1930)	355
RPM Rule Check Value/Mask 6 Register (rpm_rc_value_mask_6, Offset 0x1934)	356
RPM Rule Check Control 7 Register (rpm_rc_cntl_7, Offset 0x1938)	356
RPM Rule Check Value/Mask 7 Register (rpm_rc_value_mask_7, Offset 0x193c)	356
RPM Rule Check Control 8 Register (rpm_rc_cntl_8, Offset 0x1940)	356
RPM Rule Check Value/Mask 8 Register (rpm_rc_value_mask_8, Offset 0x1944)	356
RPM Rule Check Control 9 Register (rpm_rc_cntl_9, Offset 0x1948)	356
RPM Rule Check Value/Mask 9 Register (rpm_rc_value_mask_9, Offset 0x194c)	356
RPM Rule Check Control 10 Register (rpm_rc_cntl_10, Offset 0x1950)	357
RPM Rule Check Value/Mask 10 Register (rpm_rc_value_mask_10, Offset 0x1954)	357
RPM Rule Check Control 11 Register (rpm_rc_cntl_11, Offset 0x1958)	357
RPM Rule Check Value/Mask 11 Register (rpm_rc_value_mask_11, Offset 0x195c)	357
RPM Rule Check Control 12 Register (rpm_rc_cntl_12, Offset 0x1960)	357
RPM Rule Check Value/Mask 12 Register (rpm_rc_value_mask_12, Offset 0x1964)	358
RPM Rule Check Control 13 Register (rpm_rc_cntl_13, Offset 0x1968)	358
RPM Rule Check Value/Mask 13 Register (rpm_rc_value_mask_13, Offset 0x196c)	358
RPM Rule Check Control 14 Register (rpm_rc_cntl_14, Offset 0x1970)	358
RPM Rule Check Value/Mask 14 Register (rpm_rc_value_mask_14, Offset 0x1974)	358
RPM Rule Check Control 15 Register (rpm_rc_cntl_15, Offset 0x1978)	358
RPM Rule Check Value/Mask 15 Register (rpm_rc_value_mask_15, Offset 0x197c)	359
RPM Rule Check Configuration Register (rpm_rc_config, Offset 0x1980)	359
RLUP Command Register (rlup_command, Offset 0x2000)	360
RLUP Status Register (rplu_status, Offset 0x2004)	363
RLUP IP Source Address Register (rlup_ipsrc, Offset 0x2008)	364
RLUP IP Destination Address Register (rlup_ipdest, Offset 0x200c)	364
RLUP TCP Port Register (rlup_tcpport, Offset 0x2010)	365
RLUP Context ID Register (rlup_cid, Offset 0x2014)	365
RLUP Index Register (rlup_idx, Offset 0x2018)	367
RLUP Configuration Register (rlup_config, Offset 0x201c)	368
RLUP RSS Key 1 Register (rlup_rss_key1, Offset 0x2020)	370
RLUP RSS Key 2 Register (rlup_rss_key2, Offset 0x2024)	370
RLUP RSS Key 3 Register (rlup_rss_key3, Offset 0x2028)	370
RLUP RSS Key 4 Register (rlup_rss_key4, Offset 0x202c)	370
RLUP IPv6 RSS Key 5 Register (rlup_ipv6_rss_key5, Offset 0x2030)	370
RLUP IPv6 RSS Key 6 Register (rlup_ipv6_rss_key6, Offset 0x2034)	371

RLUP IPv6 RSS Key 7 Register (rlup_ipv6_rss_key7, Offset 0x2038)	371
RLUP IPv6 RSS Key 8 Register (rlup_ipv6_rss_key8, Offset 0x203c)	371
RLUP IPv6 RSS Key 9 Register (rlup_ipv6_rss_key9, Offset 0x2040)	372
RLUP IPv6 RSS Key 10 Register (rlup_ipv6_rss_key10, Offset 0x2044)	372
RLUP RSS Command Register (rlup_rss_command, offset 0x2048)	372
RLUP RSS Data Register (rlup_rss_data, offset 0x204c)	373
RLUP Free Count Register (rlup_free_count, offset 0x2074)	375
RLUP IPv6 Source Address 1 Register (rlup_ipv6_src1, offset 0x2078)	375
RLUP IPv6 Source Address 2 Register (rlup_ipv6_src2, offset 0x207c)	375
RLUP IPv6 Source Address 3 Register (rlup_ipv6_src3, offset 0x2080)	375
RLUP IPv6 Source Address 4 Register (rlup_ipv6_src4, offset 0x2084)	375
RLUP IPv6 Destination Address 1 Register (rlup_ipv6_dest1, offset 0x2088)	376
RLUP IPv6 Destination Address 2 Register (rlup_ipv6_dest2, offset 0x208c)	376
RLUP IPv6 Destination Address 3 Register (rlup_ipv6_dest3, offset 0x2090)	376
RLUP IPv6 Destination Address 4 Register (rlup_ipv6_dest4, offset 0x2094)	376
RLUP FTQ Data Registers (rlup_ftq_data[14], Offset 0x23c0)	376
RLUP FTQ Command Register (rlup_ftq_cmd, Offset 0x23f8)	376
RLUP FTQ Control Register (rlup_ftq_ctl, Offset 0x23fc)	380
Receive Virtual to Physical Command Scheduler (RV2PCS) Registers	381
RV2PCS Command Register (rv2pcs_ch_command, offset 0x2400)	381
RV2PCS Status Register (rv2pcs_ch_status, offset 0x2404)	381
Receive Virtual to Physical (RV2P) Registers	382
RV2P Command Register (rv2p_command, Offset 0x2800)	382
RV2P Configuration Register (rv2p_config, Offset 0x2808)	383
RV2P Instruction High Register (rv2p_instr_high, Offset 0x2830)	384
RV2P Instruction Low Register (rv2p_instr_low, Offset 0x2834)	385
RV2P Processor 1 Address/Command Register (rv2p_proc1_addr_cmd, Offset 0x2838)	386
RV2P Processor 2 Address/Command Register (rv2p_proc2_addr_cmd, Offset 0x283c)	386
RV2P Primary FTQ Data Registers (rv2p_pftq_data[14], Offset 0x2b40)	386
RV2P Post FTQ Command Register (rv2p_pftq_cmd, Offset 0x2b78)	387
RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)	389
RV2P Timeout FTQ Data Registers (rv2p_tftq_data[14], Offset 0x2b80)	389
RV2P Timeout FTQ Command Register (rv2p_tftq_cmd, Offset 0x2bb8)	390
RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)	392
RV2P Mailbox FTQ Data Registers (rv2p_mftq_data[14], Offset 0x2bc0)	392

RV2P Mailbox FTQ Command Register (rv2p_mftq_cmd, Offset 0x2bf8).....	393
RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bfc)	395
Receive DMA (RDMA) Registers	396
RDMA Command Register (rdma_command, Offset 0x2c00)	396
RDMA Status Register (rdma_status, Offset 0x2c04).....	396
RDMA Configuration Register (rdma_config, Offset 0x2c08)	397
RDMA FTQ Data Registers (rdma_ftq_data[14], Offset 0x2fc0).....	399
RDMA FTQ Command Register (rdma_ftq_cmd, Offset 0x2ff8).....	399
RDMA FTQ Control Register (rdma_ftq_ctl, Offset 0x2ffc)	401
RBDC Checksum Error Status Register (rbdc_cksum_error_status, Offset 0x3058)	402
RBDC DMA Error Register (rbdc_dma_error, Offset 0x305c)	402
Mailbox Queue (MQ) Registers	403
MQ Command Register (mq_command, Offset 0x3c00).....	403
MQ Status Register (mq_status, Offset 0x3c04).....	405
MQ Configuration Register (mq_config, Offset 0x3c08)	405
MQ Enqueue 1 Register (mq_enqueue1, Offset 0x3c0c)	407
MQ Enqueue 2 Register (mq_enqueue2, Offset 0x3c10)	408
MQ Bad Write Address Register (mq_bad_wr_addr, Offset 0x3c14)	408
MQ Bad Read Address Register (mq_bad_rd_addr, Offset 0x3c18).....	408
MQ Kernel Bypass Window Start Register (mq_knl_byp_wind_start, Offset 0x3c1c)	409
MQ Kernel Windows End Register (mq_knl_wind_end, Offset 0x3c20)	409
MQ Kernel Write Mask 1 Register (mq_knl_write_mask1, Offset 0x3c24)	409
MQ Kernel Transmit Mask 1 Register (mq_knl_tx_mask1, Offset 0x3c28)	410
MQ Kernel Command Mask 1 Register (mq_knl_cmd_mask1, Offset 0x3c2c)	410
MQ Kernel Conditional Enqueue Mask 1 Register (mq_knl_cond_enqueue_mask1, Offset 0x3c30)	410
MQ Kernel Receive V2P Mask 1 Register (mq_knl_rx_v2p_mask1, Offset 0x3c34).....	411
MQ Kernel Write Mask 2 Register (mq_knl_write_mask2, Offset 0x3c38)	411
MQ Kernel Transmit Mask 2 Register (mq_knl_tx_mask2, Offset 0x3c3c).....	411
MQ Kernel Command Mask 2 Register (mq_knl_cmd_mask2, Offset 0x3c40).....	411
MQ Kernel Conditional Enqueue Mask 2 Register (mq_knl_cond_enqueue_mask2, Offset 0x3c44)	412
MQ Kernel Receive V2P Mask 2 Register (mq_knl_rx_v2p_mask2, Offset 0x3c48).....	412
MQ Kernel Bypass Write Mask 1 Register (mq_knl_byp_write_mask1, Offset 0x3c4c).....	412
MQ Kernel Bypass Transmit Mask 1 Register (mq_knl_byp_tx_mask1, Offset 0x3c50).....	413
MQ Kernel Bypass Command Mask 1 Register (mq_knl_byp_cmd_mask1, Offset 0x3c54)	413
MQ Kernel Bypass Conditional Enqueue Mask 1 Register (mq_knl_byp_cond_enqueue_mask1, Offset	

0x3c58).....	413
MQ Kernel Bypass Receive V2P Mask 1 Register (mq_knl_byp_rx_v2p_mask1, Offset 0x3c5c).....	413
MQ Kernel Bypass Write Mask 2 Register (mq_knl_byp_write_mask2, Offset 0x3c60).....	413
MQ Kernel Bypass Transmit Mask 2 Register (mq_knl_byp_tx_mask2, Offset 0x3c64)	414
MQ Kernel Bypass Command Mask 2 Register (mq_knl_byp_cmd_mask2, Offset 0x3c68).....	414
MQ Kernel Bypass Conditional Enqueue Mask 2 Register (mq_knl_byp_cond_enqueue_mask2, Offset 0x3c6c)	414
MQ Kernel Bypass Receive V2P Mask 2 Register (mq_knl_byp_rx_v2p_mask2, Offset 0x3c70)	414
MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74)	414
MQ Memory Write Data 0 Register (mq_mem_wr_data0, Offset 0x3c78)	415
MQ Memory Write Data 1 Register (mq_mem_wr_data1, Offset 0x3c7c)	415
MQ Memory Write Data 2 Register (mq_mem_wr_data2, Offset 0x3c80)	415
MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84).....	416
MQ Memory Read Data 0 Register (mq_mem_rd_data0, Offset 0x3c88)	416
MQ Memory Read Data 1 Register (mq_mem_rd_data1, Offset 0x3c8c).....	416
MQ Memory Read Data 2 Register (mq_mem_rd_data2, Offset 0x3c90)	417
MQ Debug Vector Peek Register (mq_debug_vect_peek, offset 0x3c94)	417
MQ Immediate Data Buffer Configuration Register (mq_idb_cfg, offset 0x3ca0).....	418
MQ IDB Free Register (mq_idb_free, offset 0x3ca4)	421
MQ IDB State 0 Value Register (mq_idb_state0_val, offset 0x3cb0).....	421
MQ Configuration 2 Register (mq_config2, offset 0x3d00)	423
MQ Index Command Register (mq_idx_cmd, offset 0x3d04).....	424
MQ Index Data Register (mq_idx_data, offset 0x3d08)	425
MQ Scanner Command Register (mq_scnr_cmd, offset 0x3d0c)	425
MQ Scanner Data Register (mq_scnr_data, offset 0x3d10).....	426
MQ Map L2 0 Register (mq_map_l2_0, offset 0x3d20).....	426
MQ Map L2 1Register (mq_map_l2_1, offset 0x3d24).....	428
MQ Map L2 2 Register (mq_map_l2_2, offset 0x3d28).....	429
MQ Map L2 3 Register (mq_map_l2_3, offset 0x3d2c).....	430
MQ Map L2 4 Register (mq_map_l2_4, offset 0x3d30).....	431
MQ Map L2 5 Register (mq_map_l2_5, offset 0x3d34).....	432
MQ Map L4 0 Register (mq_map_l4_0, offset 0x3d40).....	433
MQ Map L4 1 Register (mq_map_l4_1, offset 0x3d44).....	434
MQ Map L4 2 Register (mq_map_l4_2, offset 0x3d48).....	435
MQ Map L4 3 Register (mq_map_l4_3, offset 0x3d4c).....	436

MQ Map L4 4 Register (mq_map_l4_4, offset 0x3d50)	437
MQ Map L4 5 Register (mq_map_l4_5, offset 0x3d54)	438
MQ Map L5 0 Register (mq_map_l5_0, offset 0x3d60)	439
MQ Map L5 1 Register (mq_map_l5_1, offset 0x3d64)	440
MQ Map L5 2 Register (mq_map_l5_2, offset 0x3d68)	441
MQ Map L5 3 Register (mq_map_l5_3, offset 0x3d6c)	442
MQ Map L5 4 Register (mq_map_l5_4, offset 0x3d70)	443
MQ Map L5 5 Register (mq_map_l5_5, offset 0x3d74)	444
CSCH FTQ Data Registers (csch_ch_ftq_data[14], Offset 0x43c0)	445
CSCH FTQ Command Register (csch_ch_ftq_cmd, Offset 0x43f8)	445
CSCH FTQ Control Register (csch_ch_ftq_ctl, Offset 0x43fc)	447
TSCH FTQ Data Registers (tsch_ftq_data[14], Offset 0x4fe0)	448
TSCH FTQ Command Register (tsch_ftq_cmd, Offset 0x4ff8)	448
TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4ffc)	449
Transmit Buffer Descriptor Read (TBDR) Registers	451
TBDR Command Register (tbdr_command, Offset 0x5000).....	451
TBDR Status Register (tbdr_status, Offset 0x5004)	452
TBDR Configuration Register (tbdr_config, Offset 0x5008)	453
TBDR FTQ Data Registers (tbdr_ftq_data[14], Offset 0x53c0).....	453
TBDR FTQ Command Register (tbdr_ftq_cmd, Offset 0x53f8)	454
TBDR FTQ Control Register (tbdr_ftq_ctl, Offset 0x53fc).....	456
Transmit DMA (TDMA) Registers	457
TDMA Command Register (tdma_command, Offset 0x5c00).....	457
TDMA Status Register (tdma_status, Offset 0x5c04)	458
TDMA Configuration Register (tdma_config, Offset 0x5c08)	459
TDMA Payload Producer Register (tdma_payload_prod, Offset 0x5c0c).....	461
TDMA DMAD FSM Register (tdma_dmad_fsm, Offset 0x5c80).....	462
TDMA DMAD Status Register (tdma_dmad_status, Offset 0x5c84).....	462
TDMA DR Interface FSM Register (tdma_dr_intf_fsm, Offset 0x5c88)	462
TDMA DR Interface Status Register (tdma_dr_intf_status, Offset 0x5c8c)	463
TDMA FTQ Data Registers (tdma_ftq_data[14], Offset 0x5fc0)	463
TDMA FTQ Command Register (tdma_ftq_cmd, Offset 0x5ff8)	463
TDMA FTQ Control Register (tdma_ftq_ctl, Offset 0x5ffc).....	465
Debug UART (DBU) Registers	467
DBU Command Register (dbu_command, Offset 0x6000)	467

DBU Status Register (dbu_status, Offset 0x6004)	467
DBU Configuration Register (dbu_config, Offset 0x6008)	468
DBU Timing Register (dbu_timing, Offset 0x600c)	468
DBU Receive Data Register (dbu_rxdata, Offset 0x6010)	469
DBU Transmit Data Register (dbu_txdata, Offset 0x6014)	469
Non-Volatile Memory (NVM) Registers	470
NVM Command Register (nvm_command, Offset 0x6400)	470
NVM Write Value Register (nvm_write, Offset 0x6408)	472
NVM Address Register (nvm_addr, Offset 0x640c)	473
NVM Read Value Register (nvm_read, Offset 0x6410)	475
NVM Configuration 1 Register (nvm_cfg1, Offset 0x6414)	476
NVM Configuration 2 Register (nvm_cfg2, Offset 0x6418)	478
NVM Configuration 3 Register (nvm_cfg3, Offset 0x641c)	478
NVM Software Arbitration Register (nvm_sw_arb, Offset 0x6420)	479
NVM Access Enable Register (nvm_access_enable, Offset 0x6424)	480
NVM Write 1 Register (nvm_write1, Offset 0x6428)	481
NVM Configuration Four Register (NVM_CFG4, Offset 0x642c)	481
NVM Reconfiguration Register (nvm_reconfig, Offset 0x6430)	482
Host Coalescing (HC) Registers	484
HC Command Register (hc_command, Offset 0x6800)	484
HC Status Register (hc_status, Offset 0x6804)	485
HC Configuration Register (hc_config, Offset 0x6808)	487
HC Attention Bits Enable Register (hc_attn_bits_enable, Offset 0x680c)	490
HC Status Address Low Register (hc_status_addr_l, Offset 0x6810)	490
HC Status Address High Register (hc_status_addr_h, Offset 0x6814)	490
HC Statistics Address Low Register (hc_statistics_addr_l, Offset 0x6818)	490
HC Statistics Address High Register (hc_statistics_addr_h, Offset 0x681c)	490
HC TX Quick Consumer Trip Pointer Register (hc_tx_quick_cons_trip, Offset 0x6820)	490
HC Kernel Completion Producer Trip Point Register (hc_comp_prod_trip, Offset 0x6824)	491
HC RX Quick Consumer Trip Point Register (hc_rx_quick_cons_trip, Offset 0x6828)	491
HC RX Ticks Register (hc_rx_ticks, Offset 0x682c)	492
HC TX Ticks Register (hc_tx_ticks, Offset 0x6830)	492
HC Kernel Completion Ticks Register (hc_com_ticks, Offset 0x6834)	493
HC Command Ticks Register (hc_cmd_ticks, Offset 0x6838)	493
HC Periodic Ticks Register (hc_periodic_ticks, Offset 0x683c)	494

HC Statistics Collect Ticks Register (hc_stat_collect_ticks, Offset 0x6840)	495
HC Statistics Ticks Register (hc_stats_ticks, Offset 0x6844)	495
HC Statistics Memory Data Register (hc_stat_mem_data, Offset 0x684c)	495
HC Coalesce Now Register (hc_coalesce_now, Offset 0x6914)	496
HC MSI-X Bit Vector Register (hc_msix_bit_vector, Offset 0x6918)	496
HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)	497
HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)	498
HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)	499
HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)	499
HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)	499
HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)	500
HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)	500
HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)	501
HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)	501
HC Status Block Configuration 2 Register (hc_sb_config_2, Offset 0x6a24)	502
HC TX Quick Consumer Trip 2 Register (hc_tx_quick_cons_trip_2, Offset 0x6a28)	503
HC Completion Producer Trip 2 Register (hc_comp_prod_trip_2, Offset 0x6a2c)	503
HC RX Quick Consumer Trip 2 Register (hc_rx_quick_cons_trip_2, Offset 0x6a30)	503
HC RX Ticks 2 Register (hc_rx_ticks_2, Offset 0x6a34)	503
HC TX Ticks 2 Register (hc_tx_ticks_2, Offset 0x6a38)	504
HC Completion Ticks 2 Register (hc_com_ticks_2, Offset 0x6a3c)	504
HC Command Ticks 2 Register (hc_cmd_ticks_2, Offset 0x6a40)	504
HC Periodic Ticks 2 Register (hc_periodic_ticks_2, Offset 0x6a44)	504
HC Status Block Configuration 3 Register (hc_sb_config_3, Offset 0x6a48)	504
HC TX Quick Consumer Trip 3 Register (hc_tx_quick_cons_trip_3, Offset 0x6a4c)	505
HC Completion Producer Trip 3 Register (hc_comp_prod_trip_3, Offset 0x6a50)	505
HC RX Quick Consumer Trip 3 Register (hc_rx_quick_cons_trip_3, Offset 0x6a54)	505
HC RX Ticks 3 Register (hc_rx_ticks_3, Offset 0x6a58)	505
HC TX Ticks 3 Register (hc_tx_ticks_3, Offset 0x6a5c)	505
HC Completion Ticks 3 Register (hc_com_ticks_3, Offset 0x6a60)	506
HC Command Ticks 3 Register (hc_cmd_ticks_3, Offset 0x6a64)	506
HC Periodic Ticks 3 Register (hc_periodic_ticks_3, Offset 0x6a68)	506
HC Status Block Configuration 4 Register (hc_sb_config_4, Offset 0x6a6c)	506
HC TX Quick Consumer Trip 4 Register (hc_tx_quick_cons_trip_4, Offset 0x6a70)	507
HC Completion Producer Trip 4 Register (hc_comp_prod_trip_4, Offset 0x6a74)	507

HC RX Quick Consumer Trip 4 Register (hc_rx_quick_cons_trip_4, Offset 0x6a78)	507
HC RX Ticks 4 Register (hc_rx_ticks_4, Offset 0x6a7c)	507
HC TX Ticks 4 Register (hc_tx_ticks_4, Offset 0x6a80)	507
HC Completion Ticks 4 Register (hc_com_ticks_4, Offset 0x6a84)	507
HC Command Ticks 4 Register (hc_cmd_ticks_4, Offset 0x6a88)	508
HC Periodic Ticks 4 Register (hc_periodic_ticks_4, Offset 0x6a8c)	508
HC Status Block Configuration 5 Register (hc_sb_config_5, Offset 0x6a90).....	508
HC TX Quick Consumer Trip 5 Register (hc_tx_quick_cons_trip_5, Offset 0x6a94)	509
HC Completion Producer Trip 5 Register (hc_comp_prod_trip_5, Offset 0x6a98)	509
HC RX Quick Consumer Trip 5 Register (hc_rx_quick_cons_trip_5, Offset 0x6a9c).....	509
HC RX Ticks 5 Register (hc_rx_ticks_5, Offset 0x6aa0).....	509
HC TX Ticks 5 Register (hc_tx_ticks_5, Offset 0x6aa4)	509
HC Completion Ticks 5 Register (hc_com_ticks_5, Offset 0x6aa8).....	509
HC Command Ticks 5 Register (hc_cmd_ticks_5, Offset 0x6aac).....	509
HC Periodic Ticks 5 Register (hc_periodic_ticks_5, Offset 0x6ab0)	510
HC Status Block Configuration 6 Register (hc_sb_config_6, Offset 0x6ab4).....	510
HC TX Quick Consumer Trip 6 Register (hc_tx_quick_cons_trip_6, Offset 0x6ab8)	510
HC Completion Producer Trip 6 Register (hc_comp_prod_trip_6, Offset 0x6abc).....	511
HC RX Quick Consumer Trip 6 Register (hc_rx_quick_cons_trip_6, Offset 0x6ac0).....	511
HC RX Ticks 6 Register (hc_rx_ticks_6, Offset 0x6ac4)	511
HC TX Ticks 6 Register (hc_tx_ticks_6, Offset 0x6ac8).....	511
HC Completion Ticks 6 Register (hc_com_ticks_6, Offset 0x6acc)	511
HC Command Ticks 6 Register (hc_cmd_ticks_6, Offset 0x6ad0)	511
HC Periodic Ticks 6 Register (hc_periodic_ticks_6, Offset 0x6ad4)	511
HC Status Block Configuration 7 Register (hc_sb_config_7, Offset 0x6ad8).....	512
HC TX Quick Consumer Trip 7 Register (hc_tx_quick_cons_trip_7, Offset 0x6adc)	512
HC Completion Producer Trip 7 Register (hc_comp_prod_trip_7, Offset 0x6ae0)	512
HC RX Quick Consumer Trip 7 Register (hc_rx_quick_cons_trip_7, Offset 0x6ae4)	513
HC RX Ticks 7 Register (hc_rx_ticks_7, Offset 0x6ae8)	513
HC TX Ticks 7 Register (hc_tx_ticks_7, Offset 0x6aec).....	513
HC Completion Ticks 7 Register (hc_com_ticks_7, Offset 0x6af0).....	513
HC Command Ticks 7 Register (hc_cmd_ticks_7, Offset 0x6af4)	513
HC Periodic Ticks 7 Register (hc_periodic_ticks_7, Offset 0x6af8)	513
HC Status Block Configuration 8 Register (hc_sb_config_8, Offset 0x6afc).....	513
HC TX Quick Consumer Trip 8 Register (hc_tx_quick_cons_trip_8, Offset 0x6b00)	514

HC Completion Producer Trip 8 Register (hc_comp_prod_trip_8, Offset 0x6b04).....	514
HC RX Quick Consumer Trip 8 Register (hc_rx_quick_cons_trip_8, Offset 0x6b08).....	514
HC RX Ticks 8 Register (hc_rx_ticks_8, Offset 0x6b0c).....	515
HC TX Ticks 8 Register (hc_tx_ticks_8, Offset 0x6b10).....	515
HC Completion Ticks 8 Register (hc_com_ticks_8, Offset 0x6b14).....	515
HC Command Ticks 8 Register (hc_cmd_ticks_8, Offset 0x6b18).....	515
HC Periodic Ticks 8 Register (hc_periodic_ticks_8, Offset 0x6b1c).....	515
Transmit Processor (TXP) Registers	516
TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000).....	516
TXP CPU State Register (txp_cpu_state, Offset 0x45004).....	517
TXP CPU Event Mask (txp_cpu_event_mask, Offset 0x45008).....	519
TXP CPU Program Counter Register (txp_cpu_program_counter, Offset 0x4501c).....	519
TXP CPU Register File Registers (txp_cpu_reg_file[32], Offset 0x45200).....	519
TXP CPU FTQ Data Registers (txp_ftq_data[14], Offset 0x453c0).....	520
TXP CPU FTQ Command Register (txp_ftq_cmd, Offset 0x453f8).....	520
TXP CPU FTQ Control Register (txp_ftq_ctl, Offset 0x453fc).....	521
TXP CPU Scratchpad Area (txp_scratch[8192], Offset 0x60000).....	522
Transmit Patch-Up (TPAT) Registers	523
TPAT CPU Mode Register (tpat_cpu_mode, Offset 0x85000).....	523
TPAT CPU State Register (tpat_cpu_state, Offset 0x85004).....	524
TPAT CPU Event Mask Register (tpat_cpu_event_mask, Offset 0x85008).....	526
TPAT CPU Program Counter Register (tpat_cpu_program_counter, Offset 0x8501c).....	526
TPAT CPU Register File Registers (tpat_cpu_reg_file[32], Offset 0x85200).....	526
TPAT FTQ Data Registers (tpat_ftq_data[14], Offset 0x853c0).....	527
TPAT FTQ Command Register (tpat_ftq_cmd, Offset 0x853f8).....	527
TPAT FTQ Control Register (tpat_ftq_ctl, Offset 0x853fc).....	529
TPAT Scratchpad Area (tpat_scratch[4096], Offset 0xa0000).....	529
TPAT Scratchpad Area (tpat_scratch[3072], Offset 0xa0000).....	530
Receive Processor (RXP) Registers	531
RXP CPU Mode Register (rxp_cpu_mode, Offset 0xc5000).....	531
RXP CPU State Register (rxp_cpu_state, Offset 0xc5004).....	532
RXP CPU Event Mask Register (rxp_cpu_event_mask, Offset 0xc5008).....	534
RXP CPU Program Counter Register (rxp_cpu_program_counter, Offset 0xc501c).....	534
RXP CPU Register File Registers (rxp_cpu_reg_file[32], Offset 0xc5200).....	534
RXP CFTQ Data Registers (rxp_cftq_data[14], Offset 0xc5380).....	535

RXP CFTQ Command Register (rxp_cftq_cmd, Offset 0xc53b8)	535
RXP CFTQ Control Register (rxp_cftq_ctl, Offset 0xc53bc).....	537
RXP FTQ Data Registers (rxp_ftq_data[14], Offset 0xc53c0).....	537
RXP FTQ Command Register (rxp_ftq_cmd, Offset 0xc53f8).....	538
RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)	539
RXP Scratchpad Area (rxp_scratch[8192], Offset 0xe0000)	540
RXP Scratchpad Area (rxp_scratch[9216], Offset 0xe0000)	541
RXP Scratchpad Area (rxp_scratch[10240], Offset 0xe0000)	541
Completion Processor (CP) Registers.....	542
COM CPU Mode Register (com_cpu_mode, Offset 0x105000).....	542
COM CPU State Register (com_cpu_state, Offset 0x105004).....	543
COM CPU Event Mask Register (com_cpu_event_mask, Offset 0x105008).....	545
COM CPU Program Counter Register (com_cpu_program_counter, Offset 0x10501c)	545
COM CPU Register File Registers (com_cpu_reg_file[32], Offset 0x105200).....	545
COM COMXQ FTQ Data Registers (com_comxq_ftq_data[14], Offset 0x105340)	546
COM COMXQ FTQ Command Register (com_comxq_ftq_cmd, Offset 0x105378)	547
COM COMXQ FTQ Control Register (com_comxq_ftq_ctl, Offset 0x10537c).....	549
COM COMTQ FTQ Data Registers (com_comtq_ftq_data[14], Offset 0x105380)	549
COM COMTQ FTQ Command Register (com_comtq_ftq_cmd, Offset 0x1053b8)	550
COM COMTQ FTQ Control Register (com_comtq_ftq_ctl, Offset 0x1053bc).....	552
COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)	552
COM COMQ FTQ Command Register (com_comq_ftq_cmd, Offset 0x1053f8).....	553
COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc)	555
COM Scratchpad Area (com_scratch[8192], Offset 0x120000)	555
COM Scratchpad Area (com_scratch[10240], Offset 0x120000)	556
Management Control Processor (MCP) Registers.....	557
MCP CPU Mode Register (mcp_cpu_mode, Offset 0x145000)	557
MCP CPU State Register (mcp_cpu_state, Offset 0x145004)	558
MCP CPU Event Mask Register (mcp_cpu_event_mask, Offset 0x145008)	560
MCP CPU Program Counter Register (mcp_cpu_program_counter, Offset 0x14501c)	560
MCP CPU Register File Registers (mcp_cpu_reg_file[32], Offset 0x145200)	560
MCP MCPQ FTQ Data Registers (mcp_mcpq_ftq_data[14], Offset 0x1453c0).....	561
MCP MCPQ FTQ Command Register (mcp_mcpq_ftq_cmd, Offset 0x1453f8)	561
MCP MCPQ FTQ Control Register (mcp_mcpq_ftq_ctl, Offset 0x1453fc).....	563
MCP Scratchpad Area (mcp_scratch[8192], Offset 0x160000).....	563

MCP Scratchpad Area (mcp_scratch[16384], Offset 0x160000)	564
Command Processor (COM) Registers	565
CP CPU Mode Register (cp_cpu_mode, Offset 0x185000)	565
CP CPU State Register (cp_cpu_state, Offset 0x185004)	566
CP CPU Event Mask Register (cp_cpu_event_mask, Offset 0x185008)	568
CP CPU Program Counter Register (cp_cpu_program_counter, Offset 0x18501c)	568
CP CPU Register File Registers (cp_cpu_reg_file[32], Offset 0x185200)	568
CP CPQ FTQ Data Registers (cp_cpq_ftq_data[14], Offset 0x1853c0)	569
CP CPQ FTQ Command Register (cp_cpq_ftq_cmd, Offset 0x1853f8)	570
CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc)	571
CP Scratchpad Area (cp_scratch[8192], Offset 0x1a0000)	573
CP Scratchpad Area (cp_scratch[10240], Offset 0x1a0000)	573
Transmit Assembler (TAS) Registers	574
TAS Command Register (tas_command, Offset 0x1c0000)	574
TAS FTQ Data Registers (tas_ftq_data[14], Offset 0x1c03c0)	575
TAS FTQ Command Register (tas_ftq_cmd, Offset 0x1c03f8)	576
TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)	578
RBUF Command Register (rbuf_command, Offset 0x200000)	579
RBUF Status 1 Register (rbuf_status1, Offset 0x200004)	581
RBUF Status 2 Register (rbuf_status2, Offset 0x200008)	581
RBUF Configuration Register (rbuf_config, Offset 0x20000c)	581
RBUF Firmware Buffer Allocation Register (rbuf_fw_buf_alloc, Offset 0x200010)	582
RBUF Firmware Buffer Free Register (rbuf_fw_buf_free, Offset 0x200014)	583
RBUF Firmware Buffer Select Register (rbuf_fw_buf_sel, Offset 0x200018)	583
RBUF Configuration 2 Register (rbuf_config2, Offset 0x20001c)	584
RBUF Configuration 3 Register (rbuf_config3, Offset 0x200020)	584
RBUF Packet Data Registers (rbuf_pkt_data[2250], Offset 0x208000)	585
RBUF Cluster List Data Registers (rbuf_clist_data[512], Offset 0x210000)	585
RBUF MBUF Data Registers (rbuf_buf_data[16384], Offset 0x220000)	585
PCI Express to GRC Interface (P2R) Block	586
P2R EPB Configuration Registers (p2r_epb_config[256], Offset 0x240000)	586
P2R Debug Registers (p2r_debug[256], Offset 0x240400)	586
P2R MDIO Address Register (p2r_mdio_addr, Offset 0x240800)	586
P2R MDIO Write Data Register (p2r_mdio_wr_data, Offset 0x240804)	587
P2R MDIO Read Data Register (p2r_mdio_rd_data, Offset 0x240808)	587

P2R Command Register (p2r_command, Offset 0x244000).....	587
Section 11: EPB Register Definitions	589
EPB Features	589
EPB Queues/Buffers.....	589
EPB Configuration Registers	590
Vendor ID Register (epb_vid, Offset 0x00).....	590
Device ID Register (epb_did, Offset 0x02)	590
Command Register (epb_cr, Offset 0x04).....	591
Status Register (epb_sr, Offset 0x06)	592
Revision ID Register (epb_revision_id, Offset 0x08).....	593
Class Code Register (epb_class_code, Offset 0x09).....	593
Header Type Register (epb_hdr_type, Offset 0x0e).....	593
Primary Bus Number (epb_bsnum, Offset 0x18).....	594
Secondary Primary Bus Number (epb_sbsnum, Offset 0x19).....	594
Subordinate Bus Number (epb_subbsnum, Offset 0x1a).....	594
Secondary Latency Timer (epb_sltr, Offset 0x1b)	594
I/O Base Register (epb_iobr, Offset 0x1c).....	595
I/O Limit Register (epb_iolr, Offset 0x1d)	595
Secondary Status Register (epb_ssrr, Offset 0x1e)	596
Memory Base Register (epb_mbr, Offset 0x20)	597
Memory Limit Register (epb_mlrr, Offset 0x22)	597
Prefetchable Memory Base Register (epb_pmbr, Offset 0x24)	598
Prefetchable Memory Limit Register (epb_pmlr, Offset 0x26).....	598
Prefetchable Memory Base Upper 32bit Register (epb_pmubr, Offset 0x28)	598
Prefetchable Memory Limit Upper 32-bit Register (epb_pmulr, Offset 0x2c)	599
Capabilities Pointer Register (epb_next_cap_34, Offset 0x34)	599
Bridge Control Register (epb_bridge_cr, Offset 0x3e)	599
PCI Express Capabilities Register (epb_pcier_cap_id, Offset 0x60).....	600
PCI Express Next Capability Pointer Register (epb_pcier_next_cap_ptr, Offset 0x61)	601
PCI Express Capabilities Register (epb_pcier_capability, Offset 0x62)	601
PCI Express Device Capabilities Register (epb_pcier_dev_capability, Offset 0x64)	601
PCI Express Device Control Register (epb_pcier_control, Offset 0x68)	603
PCI Express Device Status Register (epb_pcier_dev_status, Offset 0x6a)	604
PCI Express Link Capabilities Register (epb_pcier_link_capability, Offset 0x6c)	604

PCI Express Link Control Register (epb_pcie_link_control, Offset 0x70).....	606
PCI Express Link Status Register (epb_pcie_link_status, Offset 0x72).....	607
PCI Express Receive Resource Register (epb_pcie_rcv_resc, Offset 0x74)	607
PCI Express Link Special Error Status Register (epb_pcie_link_special_err_status, Offset 0x76)	608
Private Interconnect Bus Capability ID Register (epb_pib_cap_id, Offset 0x90).....	608
Private Interconnect Bus Next Capability Pointer Register (epb_pib_next_cap_ptr, Offset 0x91).....	608
PIB Secondary Status Register (epb_pib_sec_status, Offset 0x92).....	609
PIB Bridge Status Register (epb_pib_pri_status, Offset 0x94)	610
PCI Power Management Capability ID Register (epb_pm_cap_id, Offset 0xb0).....	610
PCI Power Management Next Capability Pointer Register (epb_pm_next_cap_ptr, Offset 0xb1)	610
PCI Power Management Capabilities (epb_pm_capability, Offset 0xb2).....	611
PCI Power Management Control/Status Register (epb_pm_csr, Offset 0xb4)	612
Power Management Data Register (epb_pm_data, Offset 0xb7)	613
PCI Express Enhanced Capability Header Register (epb_adverr_capability, Offset 0x100)	613
PCI Express Advanced Uncorrectable Error Status Register (epb_adverr_uncorr_error_status, Offset 0x104)	614
PCI Express Advanced Uncorrectable Error Mask Register (epb_adverr_uncorr_error_mask, Offset 0x108)	615
PCI Express Advanced Uncorrectable Error Severity Register (epb_adverr_uncorr_error_severity, Offset 0x108)	616
PCI Express Advanced Correctable Error Status Register (epb_adverr_corr_error_sta, Offset 0x110) ..	617
PCI Express Advanced Correctable Error Mask Register (epb_adverr_corr_error_mask, Offset 0x114)	617
PCI Express Advanced Capabilities and Control Register (epb_advcap_control, Offset 0x118)	618
PCI Express Header Log Register (epb_header_log, Offset 0x11c)	618
PIB Uncorrectable Error Status Register (epb_pib_uncorr_error_status, Offset 0x12c).....	619
PIB Uncorrectable Error Mask Register (epb_pib_uncorr_error_mask, Offset 0x130)	620
PIB Uncorrectable Error Severity Register (epb_pib_uncorr_error_sev, Offset 0x134).....	621
PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138).....	622
PIB Header Log Register (epb_pib_header_log, Offset 0x13c).....	622
Power Budgeting Enhanced Capability Header Register (epb_power_capability, Offset 0x14c)	623
Power Budgeting Data Select Register (epb_pb_ds, Offset 0x150)	624
Power Budgeting Data Register (epb_pb_dr, Offset 0x154).....	624
Power Budgeting Capability Register (epb_pb_cap, Offset 0x158)624
PCI Power Management Data 0 Register (pcipm_data_reg_0, offset 0x210)	624

PCI Power Management Data Register 1 (pcipm_data_reg_1, offset 0x214).....	625
Section 12: BCM5706C, BCM5706S, and BCM5708C Transceiver Register Summary...	
626	
MII Management Interface Register Programming	626
Register Map	626
Register Notations	629
1000BASE-T/100BASE-TX/10BASE-T Registers Descriptions	630
1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h).....	630
1000BASE-T/100BASE-TX/10BASE-T MII Status Register (Address 01h)	632
1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register (Address 02h and 03h)	634
1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h)	635
1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Register (Address 05h)	637
1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h).....	639
1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h)	640
1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h) ...	641
1000BASE-T Control Register (Address 09h)	643
1000BASE-T Status Register (Address 0Ah)	645
1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh)	647
1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h)	648
1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h)	650
1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h)	652
1000BASE-X Receive Error Counter Register (Address 12h).....	653
1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h)	653
1000BASE-X False Carrier Sense Counter Register (Address 13h).....	654
1000BASE-T/100BASE-TX/10BASE-T Packets Received with Transmit Error Codes Counter Register (Address 13h)	654
1000BASE-X Packets Received with Transmit Error Codes Counter Register (Address 13h)	655
1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h)	655
1000BASE-X Receiver NOT_OK Counter Register (Address 14h).....	656
1000BASE-T/100BASE-TX/10BASE-T Receive CRC Counter Register (Address 14h).....	656
1000BASE-X Receive CRC Counter Register (Address 14h).....	657
1000BASE-T/100BASE-TX/10BASE-T Expansion Access Register (Address 17h).....	657
1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Shadow Value Access Register (Address 18h)	
658	
1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow Value 000)....	

659	
10BASE-T Register (Address 18h, Shadow Value 001)	661
1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow Value 010). 663	
1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow Value 100) 665	
T1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow Value 111)	666
1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h).....	669
1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah).....	672
1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh)	675
1000BASE-T/100BASE-TX/10BASE-T Register 1Ch Access	676
1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow Value 00010) 677	
1000BASE-T/100BASE-TX/10BASE-T Clock Alignment Control Register (Address 1Ch, Shadow Value 00011)	679
1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow Value 00100) 680	
1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow Value 00101) 681	
1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow Value 01000). 683	
1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow Value 01001) 685	
1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down Register (Address 1Ch, Shadow Value 01010)	687
1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow Value 01101). 688	
1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow Value 01110). 690	
1000BASE-T/100BASE-TX/10BASE-T LED GPIO Control/Status Register (Address 1Ch, Shadow Value 01111)	692
1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow Value 11000).....	694
1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010).....	695
Auxiliary 1000BASE-X Control Register (Address 1Ch, Shadow Value 11011).....	697
Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow Value 11100)	700
Misc. 1000BASE-X Status Register (Address 1Ch, Shadow Value 11101).....	702
1000BASE-T/100BASE-TX/10BASE-T Autodetect Medium Register (Address 1Ch, Shadow Value 11110)	704
1000BASE-T/100BASE-TX/10BASE-T Mode Control Register (Address 1Ch, Shadow Value 11111) ...	

706	
1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0	708
1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1	709
1000BASE-T/100BASE-TX/10BASE-T Test 1 Register (Address 1Eh)	712
1000BASE-X Registers Descriptions	713
1000BASE-X MII Control Register (Address 00h).....	713
1000BASE-X MII Stauts Register (Address 01h)	715
1000BASE-X Auto-Negotiation Advertisement Register (Address 04h).....	717
SGMII Auto-Negotiation Advertisement Register (Address 04h).....	719
1000BASE-X Auto-Negotiation Link Partner Ability Register (Address 05h).....	720
SGMII Auto-Negotiation Link Partner Ability Register (Address 05h).....	722
1000BASE-X Auto-Negotiation Extended Status Register (Address 06h)	723
1000BASE-X IEEE Extended Status Register (Address 0Fh).....	724
Expansion Registers	725
Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h).....	725
Expansion Register 01h: Expansion Interrupt Status Register (Address 15h)	725
Expansion Register 02h: Expansion Interrupt Mask Register (Address 15h).....	727
Expansion Register 04h: Multicolor LED Selector Register (Address 15h).....	728
Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)	730
Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)	731
Section 13: BCM5709 and BCM5716 Transceiver Register Summary	732
MII Management Interface Register Programming	732
Register Map	732
Register Notations	736
1000BASE-T/100BASE-TX/10BASE-T Register Descriptions	737
1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h).....	737
1000BASE-T/100BASE-TX/10BASE-T MII Status Register (Address 01h)	739
1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register (Address 02h and 03h)	742
1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h)	743
1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register (Address 05h)	746
1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h).....	749
1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h)	751
1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h) ...	752

1000BASE-T Control Register (Address 09h).....	753
1000BASE-T Status Register (Address 0Ah)	755
1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh)	757
1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h).....	758
1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h)	760
1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h)	763
1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h)	764
1000BASE-T/100BASE-TX/10BASE-T Packets Received with Transmit Error Codes Counter Register (Address 13h)	765
1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h).....	766
1000BASE-T/100BASE-TX/10BASE-T Receive CRC Counter Register (Address 14h)	767
Expansion Register Access Register (Address 17h)	768
Auxiliary Control Shadow Value Access Register (Address 18h)	770
1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow 0h).....	771
10BASE-T Register (Address 18h, Shadow 1h)	773
1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow 2h).....	775
1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow 4h).....	777
1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow 7h)	779
1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h).....	781
1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah).....	784
1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh)	787
Register 1Ch Access	788
1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow 02h)	789
1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow 04h)	792
1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h)	794
1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow 08h).....	796
1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow 09h).....	798
1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down Register (Address 1Ch, Shadow 0Ah)....	802
SLED_1 Register (Address 1Ch, Shadow 0Bh).....	804
SLED_2 Register (Address 1Ch, Shadow 0Ch).....	805
1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow 0Dh)	806
1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow 0Eh).....	808
LED GPIO Control/Status Register (Address 1Ch, Shadow 0Fh).....	810
1000BASE-T Fast Link Drop Detection Register (Address 1Ch, Shadow 10h).....	811
SGMII Status Register (Address 1Ch, Shadow 15h)	812

Misc. 1000BASE-X Control 2 Register (Address 1Ch, Shadow 16h).....	814
1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h)	816
Auxiliary Control Register (Address 1Ch, Shadow 1Bh)	818
Auxiliary SGMII Status Register (Address 1Ch, Shadow 1Ch)	820
Misc. SGMII Status Register (Address 1Ch, Shadow 1Dh).....	822
Mode Control Register (Address 1Ch, Shadow 1Fh)	824
1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0.....	826
1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1	828
Test Register 1 (Address 1Eh)	831
Expansion Registers	833
Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h).....	833
Expansion Register 04h: Multicolor LED Selector Register (Address 15h).....	834
Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)	836
Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)	837
Expansion Register 07h: 100BASE-FX Far End Fault Register (Address 15h)	838
Expansion Register 09h: Channel Swapping (MDI Reverse) Register (Address 15h).....	839
Expansion Register 0Bh: Port Interrupt Status Register (Address 15h)	840
Expansion Register 10h: Cable Diagnostic Controls and Basic Status Register (Address 15h).....	841
Expansion Register 11h: Cable Diagnostic Results Register (Address 15h)	842
Expansion Register 12h: Cable Diagnostic Lengths Register (Address 15h)	843
Expansion Register 13h: Cable Diagnostic Lengths Register (Address 15h)	843
Expansion Register 42h: Operating Mode Status Register (Address 15h)	844
Expansion Register 44h: SGMII Lineside/Loopback Control Register (Address 15h).....	845
Expansion Register 51h: SGOUT \pm Control Register (Address 15h).....	846
Expansion Register 52h: SGOUT \pm and SCLK \pm Control Register (Address 15h)	847
Expansion Register 67h: 1000BASE-X Auto-Neg Misc. RX Status Register (Address 15h)	848
Expansion Register 70h: Soft Reset Register (Address 15h).....	849
Expansion Register 71h: Serial LED Control 1 Register (Address 15h).....	850
Expansion Register 72h: Serial LED Control 2 Register (Address 15h).....	852
Expansion Register 7Bh: BSC Control Register (Address 15h)	854
Expansion Register 7Ch: BSC Status Register (Address 15h)	856

LIST OF FIGURES

Figure 1: Broadcom BCM5706 Block Diagram.....	15
Figure 2: Broadcom BCM5708 Block Diagram.....	16
Figure 3: Broadcom BCM5709 and BCM5716 Block Diagram.....	17
Figure 4: Layer Comparison Between Standard NIC and a TCP/IP Offload NIC.....	18
Figure 5: Data Flow From Application Through Wire	21
Figure 6: Traditional DMA.....	22
Figure 7: RDMA.....	23
Figure 8: iSCSI Layers	24
Figure 9: iSCSI Ethernet Frame Encapsulation.....	25
Figure 10: iSCSI Architecture	26
Figure 11: Virtual Address vs. Physical Address Views	27
Figure 12: Free Page Chain	29
Figure 13: Chain Format with Multiple Pages.....	30
Figure 14: Host Memory Map	32
Figure 15: Internal Processor Address Space.....	33
Figure 16: NVRAM Memory Map	35
Figure 17: Big-Endian Versus Little-Endian Data Representation.....	128
Figure 18: Byte-Swapping Versus Word-Swapping	130
Figure 19: Byte-Swapping and Word-Swapping Example.....	131

LIST OF TABLES

Table 1: Feature Comparison	7
Table 2: Broadcom NetXtreme II Revision Levels	8
Table 3: Acronyms, Abbreviations, and Terms	10
Table 4: Internal Processor Scratchpad Memory Sizes	34
Table 5: Bootstrap Format	36
Table 6: Code Directory Region.....	36
Table 7: Manufacturing Information Region.....	38
Table 8: Feature Configuration Information Region.....	42
Table 9: Vital Product Data Information	48
Table 10: Licensing Information	49
Table 11: Program Image Region.....	51
Table 12: Host Status Block (Status)	73
Table 13: Status Block for MSI-X (status_msix)	73
Table 14: Host Status Block Attention Bits (status_attn_bits)	75
Table 15: Per Processor Status Block (status_per_cpu_block).....	80
Table 16: Host Statistics Block (Stat).....	81
Table 17: Receive Buffer Descriptor Chain Record (rx_bd).....	90
Table 18: Flags Field (rx_bd_flags)	90
Table 19: Transmit Buffer Descriptor Chain Record (tx_bd)	91
Table 20: Flags Field (tx_bd_flags)	91
Table 21: RX Packet L2 Frame Status Header (l2_fhdr)	92
Table 22: Errors Field (l2_fhdr_errors).....	92
Table 23: Status Field (l2_fhdr_status)	93
Table 24: L2 BD Chain Context (l2_bd_chain_context)	95
Table 25: Context Type (ctx_type)	95
Table 26: Neighbor State Context.....	96
Table 27: Context Type (type)	97
Table 28: Context Flags (flags)	97
Table 29: L2 Context (l2_context)	98
Table 30: L2 Context (l2_context)	98
Table 31: type	100
Table 32: oubits	100
Table 33: tcp_timer1	101

Table 34: tcp_timer2.....	101
Table 35: tcp_timer3.....	101
Table 36: tcp_timer4.....	102
Table 37: tcp_timer5.....	102
Table 38: tcp_flags	104
Table 39: cp_state	104
Table 40: tcp_slot_size.....	106
Table 41: tcp_tsch_xnum	106
Table 42: tcp_tx_protocol_flags.....	107
Table 43: tcp_upload_reason	108
Table 44: l4_bd_chain_v2p_proc1flags.....	108
Table 45: l4_bd_chain_v2p_flags.....	109
Table 46: RX Parser ACPI-WOL Pattern Format	112
Table 47: Procedure Descriptions	114
Table 48: Register Block Naming Convention.....	115
Table 49: Programming the Host Coalescing Registers.....	122
Table 50: Register Access Definitions.....	149
Table 51: Register Dependency Definitions	149
Table 52: Reset Forms	150
Table 53: Example Register Bit Definitions	151
Table 54: Vendor ID Register (pcicfg_vendor_id, Offset 0x00)	152
Table 55: Device ID Register (pcicfg_device_id, Offset 0x02)	152
Table 56: Command Register (pcicfg_command, Offset 0x04).....	153
Table 57: Status Register (pcicfg_status, Offset 0x06)	154
Table 58: Base Address 1 Register (pcicfg_bar_1, Offset 0x10)	156
Table 59: Base Address 2 Register (pcicfg_bar_2, Offset 0x14)	157
Table 60: Base Address 3 Register (pcicfg_bar_3, Offset 0x18)	157
Table 61: Base Address 4 Register (pcicfg_bar_4, Offset 0x14)	158
Table 62: Expansion ROM Base Address Register (pcicfg_exp_rom_bar, Offset 0x30)	159
Table 63: PCI-X Command Register (pcicfg_pcix_command, Offset 0x42).....	162
Table 64: PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)	163
Table 65: Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)	166
Table 66: Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)	167
Table 67: VPD Address/Flag Register (pcicfg_vpd_flag_addr, Offset 0x52).....	169
Table 68: MSI Control Register (pcicfg_msi_control, Offset 0x5a).....	170

Table 69: Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68)	172
Table 70: Miscellaneous Status Register (pcicfg_misc_status, Offset 0x6c)	173
Table 71: PCI Clock Control Register (pcicfg_pci_clock_control_bits, Offset 0x70)	174
Table 72: Interrupt Acknowledge Command Register (pcicfg_int_ack_cmd, Offset 0x84)	177
Table 73: MSI-X Control Register (pcicfg_msix_control, Offset 0xa2)	179
Table 74: MSI-X Table Offset Base Register (pcicfg_msix_tbl_off_bir, Offset 0xa4)	180
Table 75: MSI-X PBA Offset Base Register (pcicfg_msix_pba_off_bir, Offset 0xa8)	180
Table 76: PCI Express Capabilities Register (pcicfg_PCIE_capability, Offset 0xae)	181
Table 77: PCI Express Device Capabilities Register (pcicfg_device_capability, Offset 0xb0)	181
Table 78: PCI Express Device Control Register (pcicfg_device_control, Offset 0xb4)	182
Table 79: PCI Express Device Status Register (pcicfg_device_status, Offset 0xb6)	183
Table 80: PCI Express Link Capability Register (pcicfg_link_capability, Offset 0xb8)	184
Table 81: PCI Express Link Control Register (pcicfg_link_control, Offset 0xbc)	185
Table 82: PCI Express Link Status Register (pcicfg_link_status, Offset 0xbe)	185
Table 83: PCI Express Device Capability 2 Register (pcicfg_device_capability_2, Offset 0xd0)	187
Table 84: PCI Express Device Control 2 Register (pcicfg_device_control_2, Offset 0xd4)	187
Table 85: PCI Express Link Control 2 Register (pcicfg_link_control_2, Offset 0xdc)	188
Table 86: PCI Express Link Status 2 Register (pcicfg_link_status_2, Offset 0xde)	189
Table 87: PCI Express Device Serial Number Next Capability Pointer Register (pcicfg_device_ser_num_cap_off, Offset 0x102)	191
Table 88: PCI Express Advanced Error Next Capability Pointer Register (pcicfg_adv_err_cap_off, Offset 0x112)	192
Table 89: PCI Express Uncorrectable Error Status Register (pcicfg_ucorr_err_status, Offset 0x114)	192
Table 90: PCI Express Uncorrectable Error Mask Register (pcicfg_ucorr_err_mask, Offset 0x118)	193
Table 91: PCI Express Uncorrectable Error Severity Register (pcicfg_ucorr_err_sevr, Offset 0x11c)	193
Table 92: PCI Express Correctable Error Status Register (pcicfg_corr_err_status, Offset 0x120)	194
Table 93: PCI Express Correctable Error Mask Register (pcicfg_corr_err_mask, Offset 0x124)	194
Table 94: PCI Express Advanced Error Capability Control Register (pcicfg_adv_err_cap_control, Offset 0x128)	195
Table 95: PCI Express Power Budget Net Capability Pointer Register (pcicfg_pwr_bdgt_cap_off, Offset 0x152)	196
Table 96: PCI Express Power Budget Data Select Register (pcicfg_pwr_bdgt_data_sel, Offset 0x154)	196
Table 97: PCI Express Power Budget Data Register (pcicfg_pwr_bdgt_data, Offset 0x158)	197
Table 98: PCI Express Power Budget Capability Register (pcicfg_pwr_bdgt_capability, Offset 0x15c)	197
Table 99: PCI Express Virtual Channel Next Capability Pointer (pcicfg_vc_cap_off, Offset 0x162)	198

Table 100: PCI Express Port Virtual Channel Resource Control Register (pcicfg_vc_rsrc_control, Offset 0x174)	200
Table 101: PCI GRC Window Address Register (pci_grc_window_addr, Offset 0x400)	202
Table 102: PCI Configuration 1 Register (pci_config_1, Offset 0x404)	202
Table 103: PCI Configuration 2 Register (pci_config_2, Offset 0x408)	204
Table 104: PCI Configuration 3 Register (pci_config_3, Offset 0x40c)	207
Table 105: PCI Power Management Data A Register (pci_pm_data_a, Offset 0x410)	208
Table 106: PCI Power Management Data B Register (pci_pm_data_b, Offset 0x414)	209
Table 107: PCI Expansion ROM Address Register (pci_exp_rom_addr, Offset 0x420)	210
Table 108: PCI VPD Interface Register (pci_vpd_intf, Offset 0x428)	210
Table 109: PCI VPD Address Flag Register (pci_vpd_addr_flag, 0x42e)	211
Table 110: PCI ID Value 1 Register (pci_id_val1, Offset 0x434)	211
Table 111: PCI ID Value 2 Register (pci_id_val2, Offset 0x438)	212
Table 112: PCI ID Value 3 Register (pci_id_val3, Offset 0x43c)	213
Table 113: PCI ID Value 4 Register (pci_id_val4, Offset 0x440)	214
Table 114: PCI ID Value 5 Register (pci_id_val5, Offset 0x444)	217
Table 115: PCI PCI-X Extended Status Register (pci_pcix_extended_status, Offset 0x448)	218
Table 116: PCI ID Value 6 Register (pci_id_val6, Offset 0x44c)	219
Table 117: PCI MSI Data Register (pci_msi_data, Offset 0x450)	219
Table 118: PCI Configuration Access Command Register (pci_cfg_access_cmd, Offset 0x45c)	220
Table 119: MISC Command Register (misc_command, Offset 0x800)	221
Table 120: MISC Configuration Register (misc_cfg, Offset 0x804)	224
Table 121: MISC ID Register (misc_id, Offset 0x808)	229
Table 122: MISC Enable Status Register (misc_enable_status_bits, Offset 0x80c)	230
Table 123: MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)	232
Table 124: MISC Enable Clear Register (misc_enable_clr_bits, 0x814)	234
Table 125: MISC Clock Control Register (misc_clock_control_bits, 0x818)	236
Table 126: MISC SPIO Register (misc_spio, Offset 0x81c)	240
Table 127: MISC SPIO Interrupt Register (misc_spio_int, Offset 0x820)	241
Table 128: MISC LFSR Configuration Register (misc_config_lfsr, Offset 0x824)	242
Table 129: MISC LFSR Mask Register (misc_lfsr_mask_bits, Offset 0x828)	242
Table 130: MISC Arbitration Grant 0 Register (misc_arb_gnt0, Offset 0x868)	246
Table 131: MISC Arbitration Grant 1 Register (misc_arb_gnt1, Offset 0x86c)	247
Table 132: MISC Arbitration Grant 2 Register (misc_arb_gnt2, 0x870)	248
Table 133: MISC Arbitration Grant 3 Register (misc_arb_gnt3, Offset 0x874)	249

Table 134: MISC PRBS Control Register (misc_prbs_control, Offset 0x878—BCM5706 Only)	250
Table 135: MISC PRBS Status Register (misc_prbs_status, Offset 0x87c—BCM5706 Only)	250
Table 136: MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)	251
Table 137: MISC SMBus In Register (misc_smb_in, Offset 0x884)	253
Table 138: MISC SMBus Out Register (misc_smb_out, Offset 0x888)	254
Table 139: MISC SMBus Watchdog Register (misc_smb_watchdog, Offset 0x88c)	256
Table 140: MISC SMBus Heartbeat Register (misc_smb_heartbeat, Offset 0x890)	256
Table 141: MISC SMBus Poll ASF Register (misc_smb_poll_asf, Offset 0x894)	257
Table 142: MISC SMBus Poll Legacy Register (misc_smb_poll_legacy, Offset 0x898)	257
Table 143: MISC SMBus Retransmit Register (misc_smb_retran, Offset 0x89c)	257
Table 144: MISC SMBus Timestamp Register (misc_smb_timestamp, Offset 0x8a0)	258
Table 145: MISC PERR Enable 0 Register (misc_perr_ena0, Offset 0x8a4)	258
Table 146: MISC PERR Enable 1 Register (misc_perr_ena1, Offset 0x8a8)	260
Table 147: MISC PERR Enable 2 Register (misc_perr_ena2, Offset 0x8ac)	261
Table 148: MISC Voltage Regulator Control Register (misc_vreg_control, Offset 0x8b4)	262
Table 149: MISC Final Clock Control Value Register (misc_vreg_control, Offset 0x8b8)	264
Table 150: General-Purpose Hardware Control 0 Register (gp_hw_ctl0, Offset 0x8bc)	264
Table 151: General-Purpose Hardware Control 1 Register (gp_hw_ctl1, Offset 0x8c0)	268
Table 152: MISC New Core Control Register (misc_new_core_ctl, offset 0x8c8)	269
Table 153: MISC Checksum 16 Error Register (misc_cs16_err, offset 0x8e0)	270
Table 154: MISC Dual Media Control Register (misc_dual_media_ctrl, offset 0x8ec)	271
Table 155: MISC BIST CS2 Register (misc_bist_cs2, offset 0x91c)	274
Table 156: MISC Parity Error Status Register (misc_perr_status0, offset 0x944)	275
Table 157: MISC Parity Error Status 1 Register (misc_perr_status1, offset 0x948)	276
Table 158: MISC Parity Error Status 2 Register (misc_perr_status2, offset 0x94c)	277
Table 159: DMA Command Register (dma_command, Offset 0xc00)	278
Table 160: DMA Status Register (dma_status, Offset 0xc04)	279
Table 161: DMA Configuration Register (dma_config, Offset 0xc08)	281
Table 162: DMA Blackout Register (dma_blackout, Offset 0xc0c)	283
Table 163: DMA Read Master Setting 0 Register (dma_read_master_setting_0, offset 0xc10)	285
Table 164: DMA Read Master Setting 1 Register (dma_read_master_setting_1, 0xc14)	286
Table 165: DMA Write Master Setting 0 Register (dma_write_master_setting_0, offset 0xc18)	287
Table 166: DMA Write Master Setting 1 Register (dma_write_master_setting_1, offset 0xc1c)	288
Table 167: DMA Arbiter Register (dma_arbiter, offset 0xc20)	289
Table 168: DMA Arbiter Timers Register (dma_arb_timers, offset 0xc24)	291

Table 169: CTX Command Register (ctx_command, Offset 0x1000)	292
Table 170: CTX Status Register (ctx_status, Offset 0x1004)	294
Table 171: CTX Virtual Address Register (ctx_virt_addr, Offset 0x1008)	297
Table 172: CTX Page Table Register (ctx_page_tbl, Offset 0x100c).....	297
Table 173: CTX Data Address Register (ctx_data_adr, Offset 0x1010).....	297
Table 174: CTX Lock Register (ctx_lock, Offset 0x1018).....	298
Table 175: CTX Control Register (ctx_ctrl, offset 0x101c)	300
Table 176: CTX Access Status Register (ctx_access_status, Offset 0x1040)	301
Table 177: CTX Debug Lock Status Register (ctx_dbg_lock_status, Offset 0x1044)	302
Table 178: CTX Cache Control Status Register (ctx_cache_ctrl_status, 0x1048)	302
Table 179: CTX Cache Control State Machine Status Register (ctx_cache_ctrl_sm_status, Offset 0x104c)	303
Table 180: CTX Cache Status Register (ctx_cache_status, Offset 0x1050).....	304
Table 181: CTX DMA Status Register (ctx_dma_status, Offset 0x1054).....	304
Table 182: CTX Report Status Register (ctx_rep_status, Offset 0x1058).....	306
Table 183: CTX Checksum Error Status Register (ctx_cksum_error_status, Offset 0x105c)	308
Table 184: CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080).....	308
Table 185: CTX Cache Control Register (ctx_cache_ctrl, offset 0x10c0)	310
Table 186: CTX Host Page Table Control Register (ctx_host_page_tbl_ctrl, offset 0x10c8).....	310
Table 187: CTX Host Page Table Data 0 Register (ctx_host_page_tbl_data0, offset 0x10cc).....	311
Table 188: EMAC Mode Register (emac_mode, Offset 0x1400)	312
Table 189: EMAC Status Register (emac_status, Offset 0x1404)	314
Table 190: EMAC Attention Enable Register (emac_attention_ena, Offset 0x1408)	316
Table 191: EMAC LED Register (emac_led, Offset 0x140c).....	317
Table 192: EMAC Backoff Seed Register (emac_backoff_seed, Offset 0x1498)	319
Table 193: EMAC RX MTU Size Register (emac_rx_mtu_size, Offset 0x149c)	320
Table 194: EMAC MDIO Command Register (emac_mdio_comm, Offset 0x14ac).....	321
Table 195: EMAC MDIO Status Register (emac_mdio_status, Offset 0x14b0)	323
Table 196: EMAC MDIO Mode Register (emac_mdio_mode, Offset 0x14b4)	323
Table 197: EMAC MDIO Auto Poll Status Register (emac_mdio_auto_status, Offset 0x14b8).....	324
Table 198: EMAC TX Mode Register (emac_tx_mode, Offset 0x14bc)	325
Table 199: EMAC TX Status Register (emac_tx_status, Offset 0x14c0)	326
Table 200: EMAC TX Lengths Register (emac_tx_lengths, Offset 0x14c4).....	326
Table 201: EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8)	327
Table 202: EMAC RX Status Register (emac_rx_status, Offset 0x14cc).....	329
Table 203: EMAC Checksum Error Status Register (EMAC_CKSUM_ERROR_STATUS, Offset 0x14f0) ...	329

Table 204: Transmit Rate Limit Control Register (emac_tx_rate_limit_ctrl, Offset 0x16fc)	340
Table 205: RPM Command Register (rpm_command, Offset 0x1800)	341
Table 206: RPM Status Register (rpm_status, Offset 0x1804)	341
Table 207: RPM Configuration Register (rpm_config, Offset 0x1808).....	342
Table 208: RPM Management Packet Control Register (rpm_mgmt_pkt_ctrl, offset 0x180c).....	343
Table 209: RPM VLAN Match 0 Register (rpm_vlan_match0, Offset 0x1810)	343
Table 210: RPM VLAN Match 1 Register (rpm_vlan_match1, Offset 0x1814)	344
Table 211: RPM VLAN Match 2 Register (rpm_vlan_match2, Offset 0x1818)	344
Table 212: RPM VLAN Match 3 Register (rpm_vlan_match3, Offset 0x181c)	344
Table 213: RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820).....	345
Table 214: RPM IPv6 Programmable Extension 0 Register (rpm_ipv6_programmable_extension0, offset 0x1854)	348
Table 215: RPM Rule Check Control 0 Register (rpm_rc_cntl_0, Offset 0x1900).....	351
Table 216: RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)	354
Table 217: RPM Rule Check Configuration Register (rpm_rc_config, Offset 0x1980)	359
Table 218: RLUP Command Register (rlup_command, Offset 0x2000).....	360
Table 219: RLUP Status Register (rplu_status, Offset 0x2004)	363
Table 220: RLUP TCP Port Register (rlup_tcpport, Offset 0x2010).....	365
Table 221: RLUP CID Register (rlup_cid, Offset 0x2014).....	365
Table 222: RLUP Index Register (rlup_idx, Offset 0x2018)	367
Table 223: RLUP Configuration Register (rlup_config, Offset 0x201c).....	368
Table 224: RLUP RSS Key 1 Register (rlup_rss_key1, Offset 0x2020)	370
Table 225: RLUP RSS Key 2 Register (rlup_rss_key2, Offset 0x2024)	370
Table 226: RLUP RSS Key 3 Register (rlup_rss_key3, Offset 0x2028)	370
Table 227: RLUP RSS Key 4 Register (rlup_rss_key4, Offset 0x202c).....	370
Table 228: RLUP IPv6 RSS Key 5 Register (rlup_ipv6_rss_key5, Offset 0x2030).....	370
Table 229: RLUP IPv6 RSS Key 6 Register (rlup_ipv6_rss_key6, Offset 0x2034).....	371
Table 230: RLUP IPv6 RSS Key 7 Register (rlup_ipv6_rss_key7, Offset 0x2038).....	371
Table 231: RLUP IPv6 RSS Key 8 Register (rlup_ipv6_rss_key8, Offset 0x203c).....	371
Table 232: RLUP IPv6 RSS Key 9 Register (rlup_ipv6_rss_key9, Offset 0x2040).....	372
Table 233: RLUP IPv6 RSS Key 10 Register (rlup_ipv6_rss_key10, Offset 0x2044).....	372
Table 234: RLUP RSS Command Register (rlup_rss_command, offset 0x2048).....	372
Table 235: RLUP RSS Data Register (rlup_rss_data, offset 0x204c).....	373
Table 236: RLUP Free Count Register (rlup_free_count, offset 0x2074)	375
Table 237: RLUP IPv6 Source Address 1 Register (rlup_ipv6_src1, offset 0x2078).....	375

Table 238: RLUP IPv6 Source Address 2 Register (rlup_ipv6_src2, offset 0x207c).....	375
Table 239: RLUP IPv6 Source Address 3 Register (rlup_ipv6_src3, offset 0x2080).....	375
Table 240: RLUP IPv6 Source Address 4 Register (rlup_ipv6_src4, offset 0x2084)	375
Table 241: RLUP IPv6 Destination Address 1 Register (rlup_ipv6_dest1, offset 0x2088).....	376
Table 242: RLUP IPv6 Destination Address 2 Register (rlup_ipv6_dest2, offset 0x208c).....	376
Table 243: RLUP IPv6 Destination Address 3 Register (rlup_ipv6_dest3, offset 0x2090).....	376
Table 244: RLUP IPv6 Destination Address 4 Register (rlup_ipv6_dest4, offset 0x2094).....	376
Table 245: RLUP FTQ Command Register (rlup_ftq_cmd, Offset 0x23f8)	377
Table 246: RLUP FTQ Control Register (rlup_ftq_ctl, Offset 0x23fc).....	380
Table 247: RV2PCS Command Register (rv2pcs_ch_command, offset 0x2400).....	381
Table 248: RV2PCS Status Register (rv2pcs_ch_status, offset 0x2404).....	381
Table 249: RV2P Command Register (rv2p_command, Offset 0x2800).....	382
Table 250: RV2P Configuration Register (rv2p_config, Offset 0x2808).....	383
Table 251: RV2P Instruction High Register (rv2p_instr_high, Offset 0x2830)	384
Table 252: RV2P Instruction Low Register (rv2p_instr_low, Offset 0x2834).....	385
Table 253: RV2P Processor 1 Address/Command Register (rv2p_proc1_addr_cmd, Offset 0x2838)	386
Table 254: RV2P Processor 2 Address/Command Register (rv2p_proc2_addr_cmd, Offset 0x283c)	386
Table 255: RV2P Primary FTQ Command Register (rv2p_pftq_cmd, Offset 0x2b78)	387
Table 256: RV2P Primary FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)	389
Table 257: RV2P Timeout FTQ Command Register (rv2p_tftq_cmd, Offset 0x2bb8)	390
Table 258: RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)	392
Table 259: RV2P Mailbox FTQ Command Register (rv2p_mftq_cmd, Offset 0x2bf8)	393
Table 260: RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bfc)	395
Table 261: RDMA Command Register (rdma_command, Offset 0x2c00)	396
Table 262: RDMA Status Register (rdma_status, Offset 0x2c04)	396
Table 263: RDMA Configuration Register (rdma_config, Offset 0x2c08).....	397
Table 264: RDMA FTQ Command Register (rdma_ftq_cmd, Offset 0x2ff8)	399
Table 265: RDMA FTQ Control Register (rdma_ftq_ctl, Offset 0x2ffc)	401
Table 266: RBDC Checksum Error Status Register (rbdc_cksum_error_status, Offset 0x3058)	402
Table 267: RBDC DMA Error Register (rbdc_dma_error, Offset 0x305c)	402
Table 268: MQ Command Register (mq_command, Offset 0x3c00)	403
Table 269: MQ Status Register (mq_status, Offset 0x3c04)	405
Table 270: MQ Configuration Register (mq_config, Offset 0x3c08)	405
Table 271: MQ Enqueue 1 Register (mq_enqueue1, Offset 0x3c0c).....	407
Table 272: MQ Kernel Bypass Window Start Register (mq_knl_byp_wind_start, Offset 0x3c1c).....	409

Table 273: MQ Kernel Windows End Register (mq_knl_wind_end, Offset 0x3c20).....	409
Table 274: MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74).....	414
Table 275: MQ Memory Write Data 0 Register (mq_mem_wr_data0, Offset 0x3c78).....	415
Table 276: MQ Memory Write Data 1 Register (mq_mem_wr_data1, Offset 0x3c7c).....	415
Table 277: MQ Memory Write Data 2 Register (mq_mem_wr_data2, Offset 0x3c80).....	415
Table 278: MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84)	416
Table 279: MQ Memory Read Data 0 Register (mq_mem_rd_data0, Offset 0x3c88).....	416
Table 280: MQ Memory Read Data 1 Register (mq_mem_rd_data1, Offset 0x3c8c)	416
Table 281: MQ Memory Read Data 2 Register (mq_mem_rd_data2, Offset 0x3c90)	417
Table 282: MQ Debug Vector Peek Register (mq_debug_vect_peek, offset 0x3c94).....	417
Table 283: MQ Immediate Data Buffer Configuration Register (mq_idb_cfg, offset 0x3ca0)	418
Table 284: MQ IDB Free Register (mq_idb_free, offset 0x3ca4).....	421
Table 285: MQ IDB State 0 Value Register (mq_idb_state0_val, offset 0x3cb0)	421
Table 286: MQ Configuration 2 Register (mq_config2, offset 0x3d00)	423
Table 287: MQ Index Command Register (mq_idx_cmd, offset 0x3d04)	424
Table 288: MQ Scanner Command Register (mq_scnr_cmd, offset 0x3d0c).....	425
Table 289: MQ Map L2 0 Register (mq_map_l2_0, offset 0x3d20)	426
Table 290: MQ Map L2 1 Register (mq_map_l2_1, offset 0x3d24)	428
Table 291: MQ Map L2 2 Register (mq_map_l2_2, offset 0x3d28)	429
Table 292: MQ Map L2 3 Register (mq_map_l2_3, offset 0x3d2c)	430
Table 293: MQ Map L2 4 Register (mq_map_l2_4, offset 0x3d30)	431
Table 294: MQ Map L2 5 Register (mq_map_l2_5, offset 0x3d34)	432
Table 295: MQ Map L4 0 Register (mq_map_l4_0, offset 0x3d40)	433
Table 296: MQ Map L4 1 Register (mq_map_l4_1, offset 0x3d44)	434
Table 297: MQ Map L4 2 Register (mq_map_l4_2, offset 0x3d48)	435
Table 298: MQ Map L4 3 Register (mq_map_l4_3, offset 0x3d4c)	436
Table 299: MQ Map L4 4 Register (mq_map_l4_4, offset 0x3d50)	437
Table 300: MQ Map L4 5 Register (mq_map_l4_5, offset 0x3d54)	438
Table 301: MQ Map L5 0 Register (mq_map_l5_0, offset 0x3d60)	439
Table 302: MQ Map L5 1 Register (mq_map_l5_1, offset 0x3d64)	440
Table 303: MQ Map L5 2 Register (mq_map_l5_2, offset 0x3d68)	441
Table 304: MQ Map L5 3 Register (mq_map_l5_3, offset 0x3d6c)	442
Table 305: MQ Map L5 4 Register (mq_map_l5_4, offset 0x3d70)	443
Table 306: MQ Map L5 5 Register (mq_map_l5_5, offset 0x3d74)	444
Table 307: CSCH FTQ Command Register (csch_ch_ftq_cmd, Offset 0x43f8)	445

Table 308: CSCH FTQ Control Register (csch_ch_ftq_ctl, Offset 0x43fc)	447
Table 309: TSCH FTQ Command Register (tsch_ftq_cmd, Offset 0x4ff8)	448
Table 310: TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4ffc)	449
Table 311: TBDR Command Register (tbdr_command, Offset 0x5000)	451
Table 312: TBDR Status Register (tbdr_status, Offset 0x5004)	452
Table 313: TBDR Configuration Register (tbdr_config, Offset 0x5008)	453
Table 314: TBDR FTQ Command Register (tbdr_ftq_cmd, Offset 0x53f8)	454
Table 315: TBDR FTQ Control Register (tbdr_ftq_ctl, Offset 0x53fc)	456
Table 316: TDMA Command Register (tdma_command, Offset 0x5c00)	457
Table 317: TDMA Status Register (tdma_status, Offset 0x5c04)	458
Table 318: TDMA Configuration Register (tdma_config, Offset 0x5c08)	459
Table 319: TDMA Payload Producer Register (tdma_payload_prod, Offset 0x5c0c)	461
Table 320: TDMA DMAD FSM Register (tdma_dmad_fsm, Offset 0x5c80)	462
Table 321: TDMA DMAD Status Register (tdma_dmad_status, Offset 0x5c84)	462
Table 322: TDMA DR Interface FSM Register (tdma_dr_intf_fsm, Offset 0x5c88)	462
Table 323: TDMA DR Interface Status Register (tdma_dr_intf_status, Offset 0x5c8c)	463
Table 324: TDMA FTQ Command Register (tdma_ftq_cmd, Offset 0x5ff8)	463
Table 325: TDMA FTQ Control Register (tdma_ftq_ctl, Offset 0x5ffc)	465
Table 326: DBU Command Register (dbu_command, Offset 0x6000)	467
Table 327: DBU Status Register (dbu_status, Offset 0x6004)	467
Table 328: DBU Configuration Register (dbu_config, Offset 0x6008)	468
Table 329: DBU Timing Register (dbu_timing, Offset 0x600c)	468
Table 330: DBU Receive Data Register (dbu_rxdata, Offset 0x6010)	469
Table 331: DBU Transmit Data Register (dbu_txdata, Offset 0x6014)	469
Table 332: NVM Command Register (nvm_command, Offset 0x6400)	470
Table 333: NVM Write Value Register (nvm_write, Offset 0x6408)	472
Table 334: NVM Address Register (nvm_addr, Offset 0x640c)	473
Table 335: NVM Read Value Register (nvm_read, Offset 0x6410)	475
Table 336: NVM Configuration 1 Register (nvm_cfg1, Offset 0x6414)	476
Table 337: NVM Configuration 2 Register (nvm_cfg2, Offset 0x6418)	478
Table 338: NVM Configuration 3 Register (nvm_cfg3, Offset 0x641c)	478
Table 339: NVM Software Arbitration Register (nvm_sw_arb, Offset 0x6420)	479
Table 340: NVM Access Enable Register (nvm_access_enable, Offset 0x6424)	480
Table 341: NVM Write 1 Register (nvm_write1, Offset 0x6428)	481
Table 342: NVM Configuration Four Register (NVM_CFG4, Offset 0x642c—BCM5709 Only)	481

Table 343: NVM Reconfiguration Register (nvm_reconfig, Offset 0x6430—BCM5709 Only)	482
Table 344: HC Command Register (hc_command, Offset 0x6800).....	484
Table 345: HC Status Register (hc_status, Offset 0x6804)	485
Table 346: HC Configuration Register (hc_config, Offset 0x6808)	487
Table 347: HC TX Quick Consumer Trip Pointer Register (hc_tx_quick_cons_trip, Offset 0x6820)	490
Table 348: HC Kernel Completion Producer Trip Point Register (hc_comp_prod_trip, Offset 0x6824)	491
Table 349: HC RX Quick Consumer Trip Point Register (hc_rx_quick_cons_trip, Offset 0x6828).....	491
Table 350: HC RX Ticks Register (hc_rx_ticks, Offset 0x682c).....	492
Table 351: HC TX Ticks Register (hc_tx_ticks, Offset 0x6830).....	492
Table 352: HC Kernel Completion Ticks Register (hc_com_ticks, Offset 0x6834)	493
Table 353: HC Command Ticks Register (hc_cmd_ticks, Offset 0x6838)	493
Table 354: HC Periodic Ticks Register (hc_periodic_ticks, Offset 0x683c)	494
Table 355: HC Statistics Collect Ticks Register (hc_stat_collect_ticks, Offset 0x6840)	495
Table 356: HC Statistics Ticks Register (hc_stats_ticks, Offset 0x6844)	495
Table 357: HC Coalesce Now Register (hc_coalesce_now, Offset 0x6914)	496
Table 358: HC MSI-X Bit Vector Register (hc_msix_bit_vector, Offset 0x6918)	496
Table 359: HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)	497
Table 360: HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04).....	498
Table 361: HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)	499
Table 362: HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)	499
Table 363: HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)	499
Table 364: HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)	500
Table 365: HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)	500
Table 366: HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)	501
Table 367: HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20).....	501
Table 368: HC Status Block Configuration 2 Register (hc_sb_config_2, Offset 0x6a24)	502
Table 369: HC Status Block Configuration 3 Register (hc_sb_config_3, Offset 0x6a48)	504
Table 370: HC Status Block Configuration 4 Register (hc_sb_config_4, Offset 0x6a6c)	506
Table 371: HC Status Block Configuration 5 Register (hc_sb_config_5, Offset 0x6a90)	508
Table 372: HC Status Block Configuration 6 Register (hc_sb_config_6, Offset 0x6ab4)	510
Table 373: HC Status Block Configuration 7 Register (hc_sb_config_7, Offset 0x6ad8)	512
Table 374: HC Status Block Configuration 7 Register (hc_sb_config_7, Offset 0x6ad8)	513
Table 375: TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000).....	516
Table 376: TXP CPU State Register (txp_cpu_state, Offset 0x45004)	517
Table 377: TXP CPU Event Mask (txp_cpu_event_mask, Offset 0x45008).....	519

Table 378: TXP CPU FTQ Command Register (txp_ftq_cmd, Offset 0x453f8)	520
Table 379: TXP CPU FTQ Control Register (txp_ftq_ctl, Offset 0x453fc)	521
Table 380: TPAT CPU Mode Register (tpat_cpu_mode, Offset 0x85000).....	523
Table 381: TPAT CPU State Register (tpat_cpu_state, Offset 0x85004).....	524
Table 382: TPAT CPU Event Mask Register (tpat_cpu_event_mask, Offset 0x85008).....	526
Table 383: TPAT FTQ Command Register (tpat_ftq_cmd, Offset 0x853f8).....	527
Table 384: TPAT FTQ Control Register (tpat_ftq_ctl, Offset 0x853fc)	529
Table 385: RXP CPU Mode Register (rxp_cpu_mode, Offset 0xc5000).....	531
Table 386: RXP CPU State Register (rxp_cpu_state, Offset 0xc5004).....	532
Table 387: RXP CPU Event Mask Register (rxp_cpu_event_mask, Offset 0xc5008).....	534
Table 388: RXP CFTQ Command Register (rxp_cftq_cmd, Offset 0xc53b8)	535
Table 389: RXP CFTQ Control Register (rxp_cftq_ctl, Offset 0xc53bc).....	537
Table 390: RXP FTQ Command Register (rxp_ftq_cmd, Offset 0xc53f8).....	538
Table 391: RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)	539
Table 392: COM CPU Mode Register (com_cpu_mode, Offset 0x105000).....	542
Table 393: COM CPU State Register (com_cpu_state, Offset 0x105004).....	543
Table 394: COM CPU Event Mask Register (com_cpu_event_mask, Offset 0x105008).....	545
Table 395: COM COMXQ FTQ Command Register (com_comxq_ftq_cmd, Offset 0x105378)	547
Table 396: COM COMXQ FTQ Control Register (com_comxq_ftq_ctl, Offset 0x10537c)	549
Table 397: COM COMTQ FTQ Command Register (com_comtq_ftq_cmd, Offset 0x1053b8)	550
Table 398: COM COMTQ FTQ Control Register (com_comtq_ftq_ctl, Offset 0x1053bc)	552
Table 399: COM COMQ FTQ Command Register (com_comq_ftq_cmd, Offset 0x1053f8).....	553
Table 400: COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc)	555
Table 401: MCP CPU Mode Register (mcp_cpu_mode, Offset 0x145000)	557
Table 402: MCP CPU State Register (mcp_cpu_state, Offset 0x145004)	558
Table 403: MCP CPU Event Mask register (mcp_cpu_event_mask, Offset 0x145008)	560
Table 404: MCP MCPQ FTQ Command Register (mcp_mcpq_ftq_cmd, Offset 0x1453f8)	561
Table 405: MCP MCPQ FTQ Control Register (mcp_mcpq_ftq_ctl, Offset 0x1453fc)	563
Table 406: CP CPU Mode Register (cp_cpu_mode, Offset 0x185000)	565
Table 407: CP CPU State Register (cp_cpu_state, Offset 0x185004)	566
Table 408: CP CPU Event Mask Register (cp_cpu_event_mask, Offset 0x185008)	568
Table 409: CP CPQ FTQ Command Register (cp_cpq_ftq_cmd, Offset 0x1853f8)	570
Table 410: CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc)	571
Table 411: TAS Command Register (tas_command, Offset 0x1c0000)	574
Table 412: TAS FTQ Command Register (tas_ftq_cmd, Offset 0x1c03f8)	576

Table 413: TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)	578
Table 414: RBUF Command Register (rbuf_command, Offset 0x200000).....	579
Table 415: RBUF Status 1 Register (rbuf_status1, Offset 0x200004)	581
Table 416: RBUF Status 2 Register (rbuf_status2, Offset 0x200008)	581
Table 417: RBUF Configuration Register (rbuf_config, Offset 0x20000c)	581
Table 418: RBUF Firmware Buffer Allocation Register (rbuf_fw_buf_alloc, Offset 0x200010).....	582
Table 419: RBUF Firmware Buffer Free Register (rbuf_fw_buf_free, Offset 0x200014)	583
Table 420: RBUF Firmware Buffer Select Register (rbuf_fw_buf_sel, Offset 0x200018)	583
Table 421: RBUF Configuration 2 Register (rbuf_config2, Offset 0x20001c)	584
Table 422: RBUF Configuration 3 Register (rbuf_config3, Offset 0x200020).....	584
Table 423: P2R MDIO Address Register (p2r_mdio_addr, Offset 0x240800)	586
Table 424: P2R MDIO Write Data Register (p2r_mdio_wr_data, Offset 0x240804).....	587
Table 425: P2R MDIO Read Data Register (p2r_mdio_rd_data, Offset 0x240808)	587
Table 426: P2R Command Register (p2r_command, Offset 0x244000)	587
Table 427: Vendor ID Register (epb_vid, Offset 0x00)	590
Table 428: Device ID Register (epb_did, Offset 0x02)	590
Table 429: Command Register (epb_cr, Offset 0x04)	591
Table 430: Status Register (epb_sr, Offset 0x06).....	592
Table 431: Header Type Register (epb_hdr_type, Offset 0x0e)	593
Table 432: Primary Bus Number (epb_bsnum, Offset 0x18)	594
Table 433: Secondary Primary Bus Number (epb_sbsnum, Offset 0x19)	594
Table 434: Subordinate Bus Number (epb_subbsnum, Offset 0x1a)	594
Table 435: Secondary Latency Timer (epb_sltr, Offset 0x1b)	594
Table 436: I/O Base Register (epb_iobr, Offset 0x1c)	595
Table 437: I/O Limit Register (epb_iolr, Offset 0x1d).....	595
Table 438: Secondary Status Register (epb_ssrr, Offset 0x1e).....	596
Table 439: Memory Base Register (epb_mbr, Offset 0x20).....	597
Table 440: Memory Limit Register (epb_mlrr, Offset 0x22)	597
Table 441: Prefetchable Memory Base Register (epb_pmbr, Offset 0x24)	598
Table 442: Prefetchable Memory Limit Register (epb_pmlr, Offset 0x26)	598
Table 443: Prefetchable Memory Base Upper 32bit Register (epb_pmubr, Offset 0x28).....	598
Table 444: Prefetchable Memory Limit Upper 32-bit Register (epb_pmulr, Offset 0x2c).....	599
Table 445: Bridge Control Register (epb_bridge_cr, Offset 0x3e)	599
Table 446: PCI Express Capabilities Register (epb_pcier_capability, Offset 0x62).....	601
Table 447: PCI Express Device Capabilities Register (epb_pcied_capability, Offset 0x64)	601

Table 448: PCI Express Device Control Register (epb_pcie_control, Offset 0x68)	603
Table 449: PCI Express Device Status Register (epb_pcie_dev_status, Offset 0x6a)	604
Table 450: PCI Express Link Capabilities Register (epb_pcie_link_capability, Offset 0x6c)	604
Table 451: PCI Express Link Control Register (epb_pcie_link_control, Offset 0x70)	606
Table 452: PCI Express Link Status Register (epb_pcie_link_status, Offset 0x72)	607
Table 453: PCI Express Receive Resource Register (epb_pcie_rcv_resc, Offset 0x74)	607
Table 454: PCI Express Link Special Error Status Register (epb_pcie_link_special_err_status, Offset 0x76)	608
Table 455: PIB Secondary Status Register (epb_pib_sec_status, Offset 0x92)	609
Table 456: PIB Bridge Status Register (epb_pib_pri_status, Offset 0x94)	610
Table 457: PCI Power Management Capabilities (epb_pm_capability, Offset 0xb2)	611
Table 458: PCI Power Management Control/Status Register (epb_pm_csr, Offset 0xb4)	612
Table 459: PCI Express Enhanced Capability Header Register (epb_adverr_capability, Offset 0x100)	613
Table 460: PCI Express Advanced Uncorrectable Error Status Register (epb_adverr_uncorr_error_status, Offset 0x104)	614
Table 461: PCI Express Advanced Uncorrectable Error Mask Register (epb_adverr_uncorr_error_mask, Offset 0x108)	615
Table 462: PCI Express Advanced Uncorrectable Error Severity Register (epb_adverr_uncorr_error_severity, Offset 0x108)	616
Table 463: PCI Express Advanced Correctable Error Status Register (epb_adverr_corr_error_sta, Offset 0x110)	617
Table 464: PCI Express Advanced Correctable Error Mask Register (epb_adverr_corr_error_mask, Offset 0x114)	617
Table 465: PCI Express Advanced Capabilities and Control Register (epb_advcap_control, Offset 0x118)	618
Table 466: PIB Uncorrectable Error Status Register (epb_pib_uncorr_error_status, Offset 0x12c)	619
Table 467: PIB Uncorrectable Error Mask Register (epb_pib_uncorr_error_mask, Offset 0x130)	620
Table 468: PIB Uncorrectable Error Severity Register (epb_pib_uncorr_error_sev, Offset 0x134)	621
Table 469: PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)	622
Table 470: PIB Header Log Register (epb_pib_header_log, Offset 0x13c)	622
Table 471: Power Budgeting Enhanced Capability Header Register (epb_power_capability, Offset 0x14c)	623
Table 472: Power Budgeting Data Select Register (epb_pb_ds, Offset 0x150)	624
Table 473: Power Budgeting Data Register (epb_pb_dr, Offset 0x154)	624
Table 474: Power Budgeting Capability Register (epb_pb_cap, Offset 0x158)	624
Table 475: PCI Power Management Data 0 Register (pcipm_data_reg_0)	624
Table 476: PCI Power Management Data 1 Register (pcipm_data_reg_1)	625
Table 477: Register Map	626
Table 478: 1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h)	630

Table 479: 1000BASE-T/100BASE-TX/10BASE-TMII Status Register (Address 01h).....	632
Table 480: 1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register (Addresses 02h and 03h).....	634
Table 481: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h)	635
Table 482: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register (Address 05h)	637
Table 483: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h)	639
Table 484: 1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h)	640
Table 485: 1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h)	641
Table 486: 1000BASE-T Control Register (Address 09h)	643
Table 487: 1000BASE-T Status Register (Address 0Ah)	645
Table 488: 1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh)	647
Table 489: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h).....	648
Table 490: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h)	650
Table 491: 1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h)	652
Table 492: 1000BASE-X Receive Error Counter Register (Address 12h)	653
Table 493: 1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h)	653
Table 494: 1000BASE-X False Carrier Sense Counter Register (Address 13h)	654
Table 495: 1000BASE-T/100BASE-TX/10BASE-T Transmit Error Code Counter Register (Address 13h) ...	654
Table 496: 1000BASE-X Transmit Error Code Counter Register (Address 13h).....	655
Table 497: 1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h)	655
Table 498: 1000BASE-X Receiver NOT_OK Counter Register (Address 14h)	656
Table 499: 1000BASE-T/100BASE-TX/10BASE-T CRC Counter Register (Address 14h)	656
Table 500: 1000BASE-X Receive CRC Counter Register (Address 14h)	657
Table 501: 1000BASE-T/100BASE-TX/10BASE-T Expansion Access Register (Address 17h)	657
Table 502: Expansion Register Select Values.....	658
Table 503: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Shadow Values Access Register (Address 18h)	658
Table 504: Reading Register 18h	658
Table 505: Writing Register 18h	659
Table 506: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow Value 000)	659
Table 507: 10BASE-T Register (Address 18h, Shadow Value 001).....	661
Table 508: 1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow Value 010)	663

Table 509: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow Value 100).....	665
Table 510: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow Value 111).....	666
Table 511: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h)	669
Table 512: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah)	672
Table 513: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh).....	675
Table 514: 1000BASE-T/100BASE-TX/10BASE-T Register 1Ch Shadow Values	676
Table 515: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow Value 00010)	677
Table 516: 1000BASE-T/100BASE-TX/10BASE-T Clock Alignment Control Register (Address 1Ch, Shadow Value 00011)	679
Table 517: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow Value 00100)	680
Table 518: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow Value 00101)	681
Table 519: 1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow Value 01000)	683
Table 520: 1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow Value 01001)	685
Table 521: 1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down Register (Address 1Ch, Shadow Value 01010)	687
Table 522: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow Value 01101)	688
Table 523: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow Value 01110)	690
Table 524: 1000BASE-T/100BASE-TX/10BASE-T LED GPIO Control/Status Register (Address 1Ch, Shadow Value 01111)	692
Table 525: 1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow Value 11000)	694
Table 526: 1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010)	695
Table 527: Auxiliary 1000BASE-X Control Register (Address 1Ch, Shadow Value 11011)	697
Table 528: Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow Value 11100).....	700
Table 529: Misc. 1000BASE-X Status Register (Address 1Ch, Shadow Value 11101)	702
Table 530: 1000BASE-T/100BASE-TX/10BASE-T Autodetect Medium Register (Address 1Ch, Shadow Value 11110)	704
Table 531: 1000BASE-T/100BASE-TX/10BASE-T Mode Control Register (Address 1Ch, Shadow Value 11111)	706
Table 532: 1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0 ..	708

Table 533: 1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1	709
Table 534: 1000BASE-T/100BASE-TX/10BASE-T Test 1 Register (Address 1Eh)	712
Table 535: 1000BASE-X MII Control Register (Address 00h)	713
Table 536: 1000BASE-X MII Status Register (Address 01h).....	715
Table 537: 1000BASE-X Auto-Negotiation Advertisement Register (Address 04h)	717
Table 538: SGMII Auto-Negotiation Advertisement Register (Address 04h)	719
Table 539: 1000BASE-X Auto-Negotiation Link Partner Ability Register—Base Page (Address 05h) When in GMII/RGMII/RTBI Mode	720
Table 540: 1000BASE-X Auto-Negotiation Link Partner Ability Register—Base Page (Address 05h) When in SGMII Mode	722
Table 541: 1000BASE-X Auto-Negotiation Extended Status Register (Address 06h).....	723
Table 542: 1000BASE-X IEEE Extended Status Register (Address 0Fh)	724
Table 543: Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)	725
Table 544: Expansion Register 01h: Expansion Interrupt Status Register (Address 15h).....	725
Table 545: Expansion Register 02h: Expansion Interrupt Mask Register (Address 15h)	727
Table 546: Expansion Register 04h: Multicolor LED Selector Register (Address 15h)	728
Table 547: Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h).....	730
Table 548: Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)	731
Table 549: Register Map.....	732
Table 550: 1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h)	737
Table 551: 1000BASE-T/100BASE-TX/10BASE-TMII Status Register (Address 01h)	739
Table 552: 1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register MSB (Address 02h).....	742
Table 553: 1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register LSB (Address 03h).....	742
Table 554: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h)	743
Table 555: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register (Address 05h)	746
Table 556: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h)	749
Table 557: 1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h)	751
Table 558: 1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h)	752
Table 559: 1000BASE-T Control Register (Address 09h)	753

Table 560: 1000BASE-T Status Register (Address 0Ah).....	755
Table 561: 1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh).....	757
Table 562: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h)	758
Table 563: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h)	760
Table 564: 1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h)	763
Table 565: 1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h)	764
Table 566: 1000BASE-T/100BASE-TX/10BASE-T Transmit Error Code Counter Register (Address 13h)	765
Table 567: 1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h)	766
Table 568: 1000BASE-T/100BASE-TX/10BASE-T CRC Counter Register (Address 14h)	767
Table 569: Expansion Register Access Register (Address 17h).....	768
Table 570: Expansion Register Select Values	769
Table 571: Auxiliary Control Shadow Value Access Register (Address 18h).....	770
Table 572: Reading Register 18h.....	770
Table 573: Writing Register 18h.....	770
Table 574: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow 0h)	771
Table 575: 10BASE-T Register (Address 18h, Shadow 1h)	773
Table 576: 1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow 2h)	775
Table 577: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow 4h)	777
Table 578: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow 7h)	779
Table 579: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h)	781
Table 580: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah)	784
Table 581: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh).....	787
Table 582: Register 1Ch Shadows.....	788
Table 583: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow 02h)	789

Table 584: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow 02h)	790
Table 585: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow 04h)	792
Table 586: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h)	794
Table 587: 1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow 08h)	796
Table 588: 1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow 09h)	798
Table 589: Auto Power-Down Register (Address 1Ch, Shadow 0Ah)	802
Table 590: SLED_1 Register (Address 1Ch, Shadow 0Bh)	804
Table 591: SLED_2 Register (Address 1Ch, Shadow 0Ch)	805
Table 592: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow 0Dh)	806
Table 593: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow 0Eh)	808
Table 594: LED GPIO Control/Status Register (Address 1Ch, Shadow Value 0Fh)	810
Table 595: LED GPIO Control/Status Register (Address 1Ch, Shadow Value 10h)	811
Table 596: SGMII Status Register (Address 1Ch, Shadow 15h)	812
Table 597: Misc. 1000BASE-X Control 2 Register (Address 1Ch, Shadow 16h)	814
Table 598: 1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h)	816
Table 599: Auxiliary Control Register (Address 1Ch, Shadow 1Bh)	818
Table 600: Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow 1Ch)	820
Table 601: Misc. 1000BASE-X Status Register (Address 1Ch, Shadow 1Dh)	822
Table 602: Mode Control Register (Address 1Ch, Shadow 1Fh)	824
Table 603: 1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0 ..	826
Table 604: 1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1	828
Table 605: Test Register 1 (Address 1Eh)	831
Table 606: Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)	833
Table 607: Expansion Register 04h: Multicolor LED Selector Register (Address 15h)	834
Table 608: Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)	836
Table 609: Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)	837
Table 610: Expansion Register 07h: 100BASE-FX Far End Fault Register (Address 15h)	838
Table 611: Expansion Register 09h: Channel Swapping (MDI Reverse) Register (Address 15h)	839
Table 612: Expansion Register 0Bh: Port Interrupt Status Register (Address 15h)	840

Table 613: Expansion Register 10h: Cable Diagnostic Controls and Basic Status Register (Address 15h) ..	841
Table 614: Expansion Register 11h: Cable Diagnostic Results Register (Address 15h)	842
Table 615: Expansion Register 12h: Cable Diagnostic Lengths Register (Address 15h)	843
Table 616: Expansion Register 13h: Cable Diagnostic Lengths Register (Address 15h)	843
Table 617: Expansion Register 42h: Operating Mode Status Register (Address 15h)	844
Table 618: Expansion Register 44h: SGMII Lineside/Loopback Control Register (Address 15h).....	845
Table 619: Expansion Register 51h: SGOUT \pm Control Register (Address 15h)	846
Table 620: Expansion Register 52h: SGOUT \pm and SCLK \pm Control Register (Address 15h)	847
Table 621: Expansion Register 67h: 1000BASE-X Auto-Neg Misc. RX Status Register (Address 15h)	848
Table 622: Expansion Register 70h: Soft Reset Register (Address 15h).....	849
Table 623: Expansion Register 71h: Serial LED Control 1 Register (Address 15h).....	850
Table 624: Expansion Register 72h: Serial LED Control 2 Register (Address 15h).....	852
Table 625: Expansion Register 7Bh: BSC Control Register (Address 15h)	854
Table 626: Expansion Register 7Ch: BSC Status Register (Address 15h)	856

Section 1: Introduction

The Broadcom® NetXtreme® II provides a fully integrated OSI layer 4 and layer 5 solution. NetXtreme II adds unprecedented functionality to a network controller—TCP/IP, RDMA, and iSCSI 1.0/iSER, along with a complete 10/100/1000BASE-T Gigabit Ethernet (GbE), IEEE 802.3-compliant media access controller (MAC), as well as a UTP copper physical layer transceiver solution for high-performance network applications. By itself, the Broadcom NetXtreme II provides a complete single-chip Gigabit Ethernet controller for an Ethernet network interface card (NIC) or LAN-on-Motherboard (LOM) application, providing concurrent RDMA NIC (RNIC) and iSCSI 1.0/iSER HBA functionality.

BCM5706 FEATURE DESCRIPTION

- Single-chip solution for LOM and NIC applications
 - Integrated 10BASE-T/100BASE-TX/1000BASE-T transceivers
 - 10/100/1000 triple-speed MAC
 - Host interfaces
 - PCI v2.3—32 bit/64 bit, 33 MHz/66 MHz
 - PCI-X v1.0—64 bit, 66 MHz/100 MHz/133 MHz
 - TCP offload engine
 - Full fast path TCP offload
 - Zero Copy-capable hardware
- iSCSI controller
 - Provides iSCSI data placement
 - Provides iSER (iSCSI over RDMA) data placement
 - Offloads leader and data CRC
 - Supports marker and PDU alignment framing schemes
 - Boot support
 - Multi-connection-per-session support
- RDMA controller (RNIC)
 - Supports RDMA over TCP (iWARP)—RDMAC 1.0-compliant
 - Hardware-based data placement in application buffers without CPU intervention for both user and kernel modes
- Other performance features
 - TCP, IP, UDP checksum
 - TCP segmentation
 - Adaptive interrupts
- Robust manageability
 - PXE 2.0 remote boot
 - Wake-on-LAN (WOL) power switching circuit
 - SMBus controller
 - Statistic gathering (SNMP MIB II, Ethernet-like MIB, [IEEE 802.3x, clause 30])
 - Comprehensive diagnostic and configuration software suite
 - ACPI 1.1a-compliant—multiple power modes
- Advanced network features
 - Virtual LANs (VLANs)—IEEE 802.1q VLAN tagging; support for up to 64 VLANs
 - Jumbo frames (9 KB)
 - IEEE 802.3x flow control

- Low-power CMOS design
- On-chip power circuit controller
- 3.3V I/Os
- JTAG

BCM5708C FEATURE DESCRIPTION

- Single-chip solution for LOM and NIC applications
 - Integrated 10BASE-T/100BASE-TX/1000BASE-T transceivers
 - 10/100/1000 triple-speed MAC
 - Host interfaces
- x4 Lane PCIe™ Base Specification v1.0a-compliant
- iSCSI controller
 - Provides iSCSI data placement
 - Provides iSER (iSCSI over RDMA) data placement
 - Offloads header and data CRC
 - Accelerates processing of R2T PDUs
 - Supports FIM framing
 - iSCSI boot support
 - Multi-connection-per-session support
 - Deep Command (CMD) queue
- Large on-chip memories
 - Context = 400 KB
 - Receive buffer = 64 KB
 - Processor scratch pad = 32 KB
 - UMP Receive buffer = 3 KB
 - UMP Transmit buffer = 3 KB
 - Transmit buffer = 24-KB payload buffer and 4-KB header buffer
- RDMA controller (RNIC)
 - Supports RDMA over TCP (iWARP)—RDMAC 1.0-compliant up to 512 connections (queue pairs [QPs])
 - Hardware-based data placement in application buffers without CPU intervention for both user and kernel space
- Other performance features
 - TCP, IP, UDP checksum
 - TCP segmentation
 - Adaptive interrupts
 - Receive Side Scaling (RSS)
- TCP offload engine
 - Full fast path TCP offload for up to 1024 connections
 - Zero Copy-capable hardware
- Robust manageability
 - Universal Management Port (UMP)
 - PXE 2.0 remote boot
 - Wake-on-LAN (WOL) power switching circuit
 - IPMI network pass-through capability
 - Statistic gathering (SNMP MIB II, Ethernet-like MIB, [IEEE 802.3x, clause 30])
 - Comprehensive diagnostic and configuration software suite

- Advanced network features
 - Virtual LANs (VLAN)—IEEE 802.1q VLAN tagging; support for up to 64 VLANs
 - Jumbo frames (9 KB)
 - IEEE 802.3x flow control
- Low-power CMOS design
- On-chip power circuit controller
- 3.3V I/Os
- JTAG
- EJTAG

BCM5708S FEATURE DESCRIPTION

- Single-chip solution for LOM and NIC applications
 - 1000BASE-X 1.25-Gbaud/3.125-Gbaud SerDes transceiver
 - Host interfaces
- x4 Lane PCIe Base Specification v1.0a-compliant
- iSCSI controller
 - Provides iSCSI data placement
 - Provides iSER (iSCSI over RDMA) data placement
 - Offloads header and data CRC
 - Accelerates processing of R2T PDUs
 - Supports FIM framing
 - iSCSI boot support
 - Multi-connection-per-session support
 - Deep Command (CMD) queue
- Large on-chip memories
 - Context = 400 KB
 - Receive buffer = 64 KB
 - Processor scratch pad = 32 KB
 - UMP Receive buffer = 3 KB
 - UMP Transmit buffer = 3 KB
 - Transmit buffer = 24-KB payload buffer and 4-KB header buffer
- RDMA controller (RNIC)
 - Supports RDMA over TCP (iWARP)—RDMA 1.0-compliant up to 512 connections (queue pairs [QPs])
 - Hardware-based data placement in application buffers without CPU intervention for both user and kernel space
- Other performance features
 - TCP, IP, UDP checksum
 - TCP segmentation
 - Adaptive interrupts
 - Receive Side Scaling (RSS)
- TCP offload engine
 - Full fast path TCP offload for up to 1024 connections
 - Zero Copy-capable hardware
- Robust manageability
 - UMP
 - PXE 2.0 remote boot

- Wake-on-LAN (WOL) power switching circuit
- IPMI network pass-through capability
- Statistic gathering (SNMP MIB II, Ethernet-like MIB, [IEEE 802.3x, clause 30])
- Comprehensive diagnostic and configuration software suite
- ACPI 1.1a-compliant—multiple power modes
- Advanced network features
 - Virtual LANs (VLAN)—IEEE 802.1q VLAN tagging; support for up to 64 VLANs
 - Jumbo frames (9 KB)
 - IEEE 802.3x flow control
- Low-power CMOS design
- On-chip power circuit controller
- 3.3V I/Os
- JTAG
- EJTAG

BCM5709C FEATURE DESCRIPTION

- Single-chip solution for LOM and NIC applications
 - Dual integrated 10BASE-T/100BASE-T/1000BASE-T transceivers
 - 10/100/1000 triple-speed MAC
- x4 PCIe v1.1-compliant
- x2 PCIe v2.0-ready
- iSCSI controller
 - Provides iSCSI data placement
 - Provides iSER (iSCSI over RDMA) data placement
 - Offloads header and data CRC
 - Accelerates processing of R2T PDUs
 - Supports FIM framing
 - iSCSI boot support
 - Multi-connection-per-session support
 - Deep Command (CMD) queue
- Large on-chip memories
 - Context = 3072x18 bytes
 - Receive buffer = 64 KB
 - Processor scratch pad = 32 KB
 - UMP Receive buffer = 3 KB
 - UMP Transmit buffer = 3 KB
 - Transmit buffer = 24-KB payload buffer and 8-KB header buffer
- RDMA controller (RNIC)
 - Supports RDMA over TCP (iWARP)—RDMAC 1.0-compliant up to 512 connections (queue pairs [QPs])
 - Hardware-based data placement in application buffers without CPU intervention for both user and kernel space
- Other performance features
 - TCP, IP, UDP checksum
 - TCP segmentation
 - Adaptive interrupts



- Receive Side Scaling (RSS)
- TCP offload engine
 - Full fast path TCP offload for up to 1024 connections
 - Zero Copy-capable hardware
- Robust manageability
 - Network Controller—Sideband Interface (NC-SI)
 - Legacy UMP
 - 400-KHz SMBus
 - PXE 2.0 remote boot
 - Wake-on-LAN (WOL) power switching circuit
 - IPMI network pass-through capability
 - Statistic gathering (SNMP MIB II, Ethernet-like MIB, [IEEE 802.3x, clause 30])
 - Comprehensive diagnostic and configuration software suite
 - ACPI 1.1a-compliant—multiple power modes
 - MSI, MSI-X
- Advanced network features
 - Virtual LANs (VLAN)—IEEE 802.1q VLAN tagging; support for up to 64 VLANs
 - Jumbo frames (9 KB)
 - IEEE 802.3x flow control
 - IEEE 802.3ap Ethernet Operation over Electrical Backplane
 - Remote PHY
- Low-power CMOS design
- On-chip power circuit controller
- 3.3V I/Os
- JTAG

BCM5709S FEATURE DESCRIPTION

- Single-chip solution for LOM and NIC applications
 - Dual integrated 10BASE-T/100BASE-T/1000BASE-T transceivers and dual 1000BASE-X 1.25-Gbaud/3.125-Gbaud SerDes transceivers
- x4 PCIe v1.1-compliant
- x2 PCIe v2.0-ready
- iSCSI controller
 - Provides iSCSI data placement
 - Provides iSER (iSCSI over RDMA) data placement
 - Offloads header and data CRC
 - Accelerates processing of R2T PDUs
 - Supports FIM framing
 - iSCSI boot support
 - Multi-connection-per-session support
 - Deep Command (CMD) queue
- Large on-chip memories
 - Context = 3072x18 bytes
 - Receive buffer = 64 KB
 - Processor scratch pad = 32 KB

- UMP Receive buffer = 3 KB
- UMP Transmit buffer = 3 KB
- Transmit buffer = 24-KB payload buffer and 8-KB header buffer
- RDMA controller (RNIC)
 - Supports RDMA over TCP (iWARP)—RDMA 1.0-compliant up to 512 connections (queue pairs [QPs])
 - Hardware-based data placement in application buffers without CPU intervention for both user and kernel space
- Other performance features
 - TCP, IP, UDP checksum
 - TCP segmentation
 - Adaptive interrupts
 - Receive Side Scaling (RSS)
- TCP offload engine
 - Full fast path TCP offload for up to 1024 connections
 - Zero Copy-capable hardware
- Robust manageability
 - NC-SI
 - Legacy UMP
 - 400-kHz SMBus
 - PXE 2.0 remote boot
 - Wake-on-LAN (WOL) power switching circuit
 - IPMI network pass-through capability
 - Statistic gathering (SNMP MIB II, Ethernet-like MIB, [IEEE 802.3x, clause 30])
 - Comprehensive diagnostic and configuration software suite
 - ACPI 1.1a compliant—multiple power modes
 - MSI, MSI-X
- Advanced network features
 - Virtual LANs (VLAN)—802.1q VLAN tagging; support for up to 64 VLANs
 - Jumbo frames (9 KB)
 - IEEE 802.3x flow control
 - IEEE 802.3ap Ethernet Operation over Electrical Backplane
 - Remote PHY
- Low-power CMOS design
- On-chip power circuit controller
- 3.3V I/Os
- JTAG

BCM5716 FEATURE DESCRIPTION

- Single-chip solution for LOM and NIC applications
 - Dual integrated 10BASE-T/100BASE-T/1000BASE-T transceivers
 - 10/100/1000 triple-speed MAC
- x4 PCIe v1.1-compliant
- x2 PCIe v2.0-ready
- Large on-chip memories
 - Context = 3072x18 bytes
 - Receive buffer = 64 KB

- Processor scratch pad = 32 KB
- UMP Receive buffer = 3 KB
- UMP Transmit buffer = 3 KB
- Transmit buffer = 24-KB payload buffer and 8-KB header buffer
- Other performance features
 - TCP, IP, UDP checksum
 - TCP segmentation
 - Adaptive interrupts
 - Receive Side Scaling (RSS)
- Robust manageability
 - NC-SI
 - Legacy UMP
 - 400-kHz SMBus
 - PXE 2.0 remote boot
 - Wake-on-LAN (WOL) power switching circuit
 - IPMI network pass-through capability
 - Statistic gathering (SNMP MIB II, Ethernet-like MIB, [IEEE 802.3x, clause 30])
 - Comprehensive diagnostic and configuration software suite
 - ACPI 1.1a-compliant—multiple power modes
 - MSI, MSI-X
- Advanced network features
 - Virtual LANs (VLAN)—IEEE 802.1q VLAN tagging; support for up to 64 VLANs
 - Jumbo frames (9 KB)
 - IEEE 802.3x flow control
- Low-power CMOS design
- On-chip power circuit controller
- 3.3V I/Os
- JTAG

FEATURE COMPARISON

Table 1 provides a short comparison of the major features for each member of the NetXtreme II product family. For a detailed feature list, refer to the respective data sheet for the particular controller.

Table 1: Feature Comparison

Feature	BCM5706C	BCM5706S	BCM5708C	BCM5708S	BCM5709C	BCM5709S	BCM5716
Bus Interfaces							
PCI-X v1.0 64-bit 66 MHz/ 100 MHz/133 MHz	Yes	Yes	No	No	No	No	No
PCI Express v1.1 x4	No	No	Yes ^a	Yes ^a	Yes	Yes	Yes
PCI Express v2.0 x4	No	No	No	No	Yes	Yes	Yes
LAN Interfaces							
Integrated 10/100/1000BASE-T Integrated Transceiver	Yes	No	Yes	No	Yes	Yes	Yes

Table 1: Feature Comparison (Cont.)

Feature	BCM5706C	BCM5706S	BCM5708C	BCM5708S	BCM5709C	BCM5709S	BCM5716
1000BASE-X Integrated Transceiver	No	Yes	No	No	No	No	No
1000/2500BASE-X Integrated Transceiver	No	No	No	Yes	No	Yes	No
Data Management							
IEEE 802.1q VLAN Tag Support	Yes						
IEEE 802.1p L2 Priority Encoding	Yes						
IEEE 802.3ad Link Aggregation	Yes						
Frame/Packet Buffer Memory	64-KB RX 28-KB TX	64-KB RX 28-KB TX	64-KB RX 28-KB TX	64-KB RX 28-KB TX	64-KB RX 32-KB TX	64-KB RX 32-KB TX	64-KB RX 32-KB TX
Jumbo Frame Support (9 KB)	Yes						
IPv4 Checksum Offload	Yes						
IPv6 Checksum Offload	No	No	No	No	Yes	Yes	Yes
TCP Checksum Offload	Yes						
UDP Checksum Offload	Yes						
Large Send Offload (TSO/LSO)	Yes						
TCP Offload Engine (TOE)	Yes	Yes	Yes	Yes	Yes	Yes	No
iSCSI Offload	Yes	Yes	Yes	Yes	Yes	Yes	No
Manageability Support							
IPMI Support	Yes						
UMP Support	Yes						
NC-SI Support	No	No	No	No	Yes	Yes	Yes

a. The BCM5708 controller includes both a PCIe to PCI-X bridge (EPB) as well as a PCI-X Ethernet MAC core.

REVISION LEVELS

Table 2 lists the various revision levels of the Broadcom NetXtreme II. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the Broadcom NetXtreme II device, and to load the appropriate workarounds described in the corresponding errata sheets. Refer to the appropriate product data sheets for exterior part marking information. The Broadcom PCI Vendor ID is 0x14e4.

Table 2: Broadcom NetXtreme II Revision Levels

Family Member	Device ID^a	Revision Level	PCI Revision ID^b	Chip ID^c	PHY Core	Errata
BCM5706C	0x164a	A0	0x00	0x6000	BCM5464 A3	5706C-ES10X-R
BCM5706C	0x164a	A1	0x01	0x6001	BCM5464 A3	5706C-ES20X-R
BCM5706C	0x164a	A2	0x02	0x6002	BCM5464 A3	5706C-ES30X-R



Table 2: Broadcom NetXtreme II Revision Levels (Cont.)

Family Member	Device ID^a	Revision Level	PCI Revision ID^b	Chip ID^c	PHY Core	Errata
BCM5706C	0x164a	A3	0x03	0x6003	BCM5464 A3	5706C-ES40X-R
BCM5706S	0x16aa	A0	0x00	0x6000	BCM5464 A3	—
BCM5706S	0x16aa	A1	0x01	0x6001	BCM5464 A3	5706S-ES10X-R
BCM5706S	0x16aa	A2	0x02	0x6002	BCM5464 A3	5706S-ES20X-R
BCM5706S	0x16aa	A3	0x03	0x6003	BCM5464 A3	5706S-ES30X-R
BCM5708C	0x164c	A0	0x00	0x8000	BCM5464 A3	5708C-ES10X-R
BCM5708C	0x164c	B0	0x10	0x8100	BCM5464 A3	5708C-ES20X-R
BCM5708C	0x164c	B1	0x11	0x8101	BCM5464 A3	5708C-ES3XX-R
BCM5708C	0x164c	B2	0x12	0x8102	BCM5464 A3	5708C-ES4XX-R
BCM5708S	0x16ac	A0	0x00	0x8000	N/A	5708S-ES10X-R
BCM5708S	0x16ac	B0	0x10	0x8100	N/A	5708S-ES20X-R
BCM5708S	0x16ac	B1	0x11	0x8101	N/A	5708S-ES3XX-R
BCM5708S	0x16ac	B2	0x12	0x8102	N/A	5708S-ES4XX-R
BCM5709C	0x1639	A0	0x00	0x9000	BCM54980	5709C-ES10X-R
BCM5709C	0x1639	A1	0x01	0x9001	BCM54980	5709C-ES10X-R
BCM5709C	0x1639	B0	0x10	0x9100	BCM54980	5709C-ES10X-R
BCM5709C	0x1639	B1	0x11	0x9101	BCM54980	5709C-ES10X-R
BCM5709C	0x1639	B2	0x12	0x9102	BCM54980	5709C-ES10X-R
BCM5709C	0x1639	C0	0x20	0x9200	BCM54980	5709C-ES10X-R
BCM5709S	0x163a	A0	0x00	0x9000	BCM54980	5709S-ES10X-R
BCM5709S	0x163a	A1	0x01	0x9001	BCM54980	5709S-ES10X-R
BCM5709S	0x163a	B0	0x10	0x9100	BCM54980	5709S-ES10X-R
BCM5709S	0x163a	B1	0x11	0x9101	BCM54980	5709S-ES10X-R
BCM5709S	0x163a	B2	0x12	0x9102	BCM54980	5709S-ES10X-R
BCM5709S	0x163a	C0	0x20	0x9200	BCM54980	5709S-ES10X-R
BCM5716	0x163b	C0	0x20	0x9200	BCM54980	5716-ES10X-R

a. See “Device ID Register (pcicfg_device_id, Offset 0x02)” on page 152.

b. See “Revision ID Register (pcicfg_revision_id, Offset 0x08)” on page 155.

c. See “Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68)” on page 172.



Note: The devices BCM5706, BCM5708, BCM5709, and BCM5716 may also be referred to as 06, 08, 09, and 16, respectively, throughout this document.

NOTATIONAL CONVENTIONS

SIZES

- Byte = 8 bits
- Word = 32 bits
- kbit = Kilobit (1024 bits)

- KB = Kilobyte (1024 bytes)
- Mbit = Megabit (1048576 bits)
- MB = Megabyte (1048576 bytes)

ACRONYMS, ABBREVIATIONS, AND TERMS

Table 3 lists the acronyms, abbreviations, and terms used in this document.

Table 3: Acronyms, Abbreviations, and Terms

Item	Definition
ASF	Alert Specification Forum
BD	Buffer Descriptor
CAM	Content Addressable Memory
CDB	Control Data Blocks
CID	Context ID
COM	Completion Processor
CP	Command Processor
DA	Destination MAC Address
FIM	Fixed Interval Marker
FSM	Finite State Machine
FTQ	Flow-Through Queue. An FTQ is queue of data descriptors, implemented as a hardware FIFO, which allows the various state machines to operate independently of their neighbors.
GRC	Global Register Controller
HC	Host Coalescing
Header Alignment	An attribute guaranteed by the sender and verified by the receiver where the ULP's PDU is confined within a TCP segment and immediately follows the TCP header
HPC	High-performance computing
IDB	Immediate Data Buffer
IETF	Internet Engineering Task Force
IMD	Intelligent Management Device
IOPS	I/O Operations Per Second
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
iSCSI	Internet SCSI. An IETF standard for carrying SCSI commands over IP networks.
ISR	Interrupt Service Routine
iWARP	A suite of wire protocols comprised of [RDMPA], [DDP], and [MPA] for [TCP], or [RDMA] and [DDP] for [STCP].
L2	Protocols that are responsible for delivery of data between two points on a physical network, L2 protocols do not provide routing across multiple physical networks. Example: Ethernet IEEE 802.3 standard.
L3	Protocols that provide routing across multiple networks, each of which may use a different L2 protocol. L3 protocols do not provide connection management or flow control of any kind. Example: IPv4 and IPv6.
L4	Protocols that provide connection management and flow control between two systems or applications, but it has no concept of the nature of the traffic that is passing over it. Example: TCP and UDP.
L5	Protocol that maps the ULP that an application communicates to an L4 level. In doing so, it may break the application's requests into manageable PDUs. Example: iSCSI 1.0 and RDMA are examples of L5 protocols.
LLP	Lower Layer Protocol
LOM	LAN-on-Motherboard
LSO	Large Send Offload
MAC	Media Access Controller

Table 3: Acronyms, Abbreviations, and Terms (Cont.)

Item	Definition
MBA	Multiple Boot Agent
MCP	Management Processor
MSI	Message Signaled Interrupts
MSS	Maximum Segment Size
NIC	Network Interface Card
NOS	Network Operating System
PDU	Protocol Data Unit. The PDU is the complete L5 protocol item that contains ULP data and control to describe where the data is to be placed. Each PDU has its own header (which may span multiple Ethernet packets).
PHY	Physical Interface
PXE	Preboot Execution Environment
QOS	Quality of Service
RCB	Ring Control Block
RDMA	Remote Direct Memory Access
RDMAC	RDMA Consortium
RDMAP	Remote DMA protocol
RNIC	RDMA NIC
RSS	Receive Side Scaling
RXP	RX Processor
SA	Source MAC Address
SACK	Selective acknowledgement
SCSI	Small Computer System Interface
SDP	Sockets-direct protocol
SMB	System Management Bus
TCP	Transmission Control Protocol
TCP Tuple	A tuple consisting of the: <ul style="list-style-type: none">• Source IP address• Destination IP address• Source TCP port• Destination TCP port• Protocol = TCP
TOE	TCP Offload Engine. A TOE adapter allows existing TCP network connections to be moved to and from a TCP/IP stack implemented in the TOE adapter's hardware, freeing the host CPU for other tasks.
TOS	Type of Service
TPAT	TX Patchup Processor
TTL	Time-to-Live
TXP	TX Processor
Tuple	An ordered set of values.
UDP	User Datagram Protocol
ULP	Upper Layer Protocol. The ULP is the protocol implemented by two applications to allow them to communicate over some network.
UMB	Upper Memory Block
UMP	Universal Management Port
VIA	Virtual Interface Architecture
VLAN	Virtual LAN
VPD	Vital Product Data
WOL	Wake-on-LAN
WSD	Winsock Direct Protocol

RELATED DOCUMENTATION

- (ASF) *Alerting Standard Forum Specification*, v1.03, <http://www.dmtf.org/var/release/ASF/DSP0114.pdf>.
- (IBTA) *Infiniband*, <http://www.infiniband.org>.
- (IPMI) *Intelligent Platform Management Interface (IPMI) Specification* v1.5, <http://developer.intel.com/design/servers/ipmi/>.
- (PCI) *PCI Local Bus Specification*, revision 2.3, http://www.pcisig.com/specifications/conventional_pci_23.
- (PCIX) *PCI-X Addendum to the PCI Local Bus Specification*, revision 1.0a, http://www.pcisig.com/specifications/pci_x.
- (PCI_PM) *PCI Bus Power Management Interface Specification*, revision 1.1, http://www.pcisig.com/specifications/pci_bus_power_management_interface.
- (PCI_HP) *PCI Hot Plug Specification*, revision 1.1, http://www.pcisig.com/specifications/pci_hot_plug.
- (SMB) *System Management Bus Specification*, v2.0, <http://www.smbus.org/specs/index.html>.
- (PCI_SMB) *PCI ECR for Addition of the SMBus Interface to Connector*, http://www.pcisig.com/data/specifications/smb_ecn_040501.pdf.
- (RFC0768) *User Datagram Protocol (UDP)*, RFC 768, <http://www.ietf.org/rfc/rfc0768.txt>.
- (RFC0791) *Internet Protocol (IP)*, RFC 791, <http://www.ietf.org/rfc/rfc0791.txt>.
- (RFC0793) *Transmission Control Protocol (TCP)*, RFC 793, <http://www.ietf.org/rfc/rfc0793.txt>.
- (RFC0894) *A Standard for the Transmission of IP Datagrams Over Ethernet Networks*, RFC 894, <http://www.ietf.org/rfc/rfc0894.txt>.
- (RFC1042) *A Standard for the Transmission of IP Datagrams Over IEEE 802 Networks*, RFC 1042, <http://www.ietf.org/rfc/rfc1042.txt>.
- (RFC1122) *Requirements for Internet Hosts—Communication Layers*, RFC 1122, <http://www.ietf.org/rfc/rfc1122.txt>.
- (RFC1156) *Management Information Base (MIB) for Network Management of TCP/IP-based Internets*, RFC 1156, <http://www.ietf.org/rfc/rfc1156.txt>.
- (RFC1157) *A Simple Network Management Protocol (SNMP)*, RFC 1157, <http://www.ietf.org/rfc/rfc1157.txt>.
- (RFC1213) *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*, RFC 1213, <http://www.ietf.org/rfc/rfc1213.txt>.
- (RFC1573) *Evolution of the Interfaces Group of MIB-II*, RFC 1573, <http://www.ietf.org/rfc/rfc1573.txt>.
- (RFC1643) *Definitions of Managed Objects for the Ethernet-like Interface Types*, RFC 1643, <http://www.ietf.org/rfc/rfc1643.txt>.
- (RFC1700) *Assigned Numbers*, RFC 1700, <http://www.ietf.org/rfc/rfc1700.txt>.
- (RFC1750) *Randomness Recommendations for Security*, RFC 1750, <http://www.ietf.org/rfc/rfc1750.txt>.
- (RFC1757) *Remote Network Monitoring Management Information Base*, RFC 1757, <http://www.ietf.org/rfc/rfc1757.txt>.
- (RFC1828) *IP Authentication Using Keyed MD5*, RFC 1828, <http://www.ietf.org/rfc/rfc1828.txt>.
- (RFC1829) *The ESP DES-CBC Transform*, RFC 1829, <http://www.ietf.org/rfc/rfc1829.txt>.
- (RFC2011) *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2*, RFC2011, <http://www.ietf.org/rfc/rfc2011.txt>.
- (RFC2085) *HMAC-MD5 IP Authentication with Replay Prevention*, RFC 2085, <http://www.ietf.org/rfc/rfc2085.txt>.
- (RFC2104) *HMAC: Keyed-Hashing for Message Authentication*, RFC 2104, <http://www.ietf.org/rfc/rfc2104.txt>.
- (RFC2401) *Security Architecture for the Internet Protocol*, RFC 2401, <http://www.ietf.org/rfc/rfc2401.txt>.
- (RFC2402) *IP Authentication Header*, RFC 2402, <http://www.ietf.org/rfc/rfc2402.txt>.
- (RFC2403) *The Use of HMAC-MD5-96 within ESP and AH*, RFC 2403, <http://www.ietf.org/rfc/rfc2403.txt>.
- (RFC2404) *The Use of HMAC-SHA-1-96 within ESP and AH*, RFC 2404, <http://www.ietf.org/rfc/rfc2404.txt>.
- (RFC2405) *The ESP DES-CBC Cipher Algorithm with Explicit IV*, RFC 2405, <http://www.ietf.org/rfc/rfc2405.txt>.
- (RFC2406) *IP Encapsulating Security Payload (ESP)*, RFC 2406, <http://www.ietf.org/rfc/rfc2406.txt>.
- (RFC2407) *The Internet IP Security Domain of Interpretation for ISAKMP*, RFC 2407, <http://www.ietf.org/rfc/rfc2407.txt>.



- (RFC2408) *Internet Security Association and Key Management Protocol (ISAKMP)*, RFC 2408, <http://www.ietf.org/rfc/rfc2408.txt>.
- (RFC2409) *The Internet Key Exchange (IKE)*, RFC 2409, <http://www.ietf.org/rfc/rfc2409.txt>.
- (RFC2410) *The NULL Encryption Algorithm and its Use with IPSEC*, RFC 2410, <http://www.ietf.org/rfc/rfc2410.txt>.
- (RFC2411) *IP Security Document Roadmap*, RFC 2411, <http://www.ietf.org/rfc/rfc2411.txt>.
- (RFC2412) *The OAKLEY Key Determination Protocol*, RFC 2412, <http://www.ietf.org/rfc/rfc2412.txt>.
- (RFC2451) *The ESP CBC-Mode Cipher Algorithms*, RFC 2451, <http://www.ietf.org/rfc/rfc2451.txt>.
- (RFC2460) *Internet Protocol Version 6 (Ipv6)*, RFC 2460, <http://www.ietf.org/rfc/rfc2460.txt>.
- (RFC2464) *Transmission of IPv6 Packets Over Ethernet Networks*, RFC 2464, <http://www.ietf.org/rfc/rfc2464.txt>.
- [RFC2472] *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, RFC 2474, <http://www.ietf.org/rfc/rfc2474.txt>.
- [RFC2525] *Known TCP Implementation Problems*, RFC 2525, <http://www.ietf.org/rfc/rfc2525.txt>.
- [RFC2581] *TCP Congestion Control*, RFC 2581, <http://www.ietf.org/rfc/rfc2581.txt>.
- [RFC2582] *The NewReno Modification to TCP's Fast Recovery Algorithm*, RFC 2582, <http://www.ietf.org/rfc/rfc2582.txt>.
- [RFC2988] *Computing TCP's Retransmission Timer*, RFC 2988, <http://www.ietf.org/rfc/rfc2988.txt>.
- [RFC2018] *TCP Selective Acknowledgement Options*, RFC 2018, <http://www.ietf.org/rfc/rfc2018.txt>.
- [RFC2883] *An Extension to the Selective Acknowledgement (SACK) Option for TCP*, RFC 2883, <http://www.ietf.org/rfc/rfc2883.txt>.
- [RFC3517] *A Conservative Select Acknowledgement (SACK) Based Loss Recovery Algorithm for TCP*, RFC 3517, <http://www.ietf.org/rfc/rfc3517.txt>.
- [RFC3347] *Small Computer Systems Interface Protocol Over the Internet (iSCSI) Requirements and Design Considerations*, RFC 3347, <http://www.ietf.org/rfc/rfc3347.txt>.
- [RFC3385] *Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC)/Checksum Considerations*, RFC 3385, <http://www.ietf.org/rfc/rfc3385.txt>.
- [RFC3720] *Internet Small Computer Systems Interface (iSCSI)*, RFC 3720, <http://www.ietf.org/rfc/rfc3720.txt>.
- [RFC3721] *Internet Small Computer Systems Interface (iSCSI) Naming and Discovery*, RFC 3721, <http://www.ietf.org/rfc/rfc3721.txt>.
- (VIA) *Virtual Interface Architecture*, http://developer.intel.com/design/servers/vi/the_spec/specification.htm.

Section 2: Hardware Architecture

THEORY OF OPERATIONS

The Broadcom NetXtreme II includes a 10/100/1000 Ethernet MAC. There are two packet flows in this MAC—transmit and receive. The device's DMA engine will bus-master packets from host memory to device local storage, and vice-versa. The RX MAC moves packets from the PHY into internal memory. All incoming packets are checked against a set of QOS rules and then categorized. When a packet is transmitted, the TX MAC moves the data from internal memory to the PHY: both flows operate independently of each other in full-duplex mode. Six RISC processors are implemented to allow field upgrades for functionality such as TOE and iSCSI.

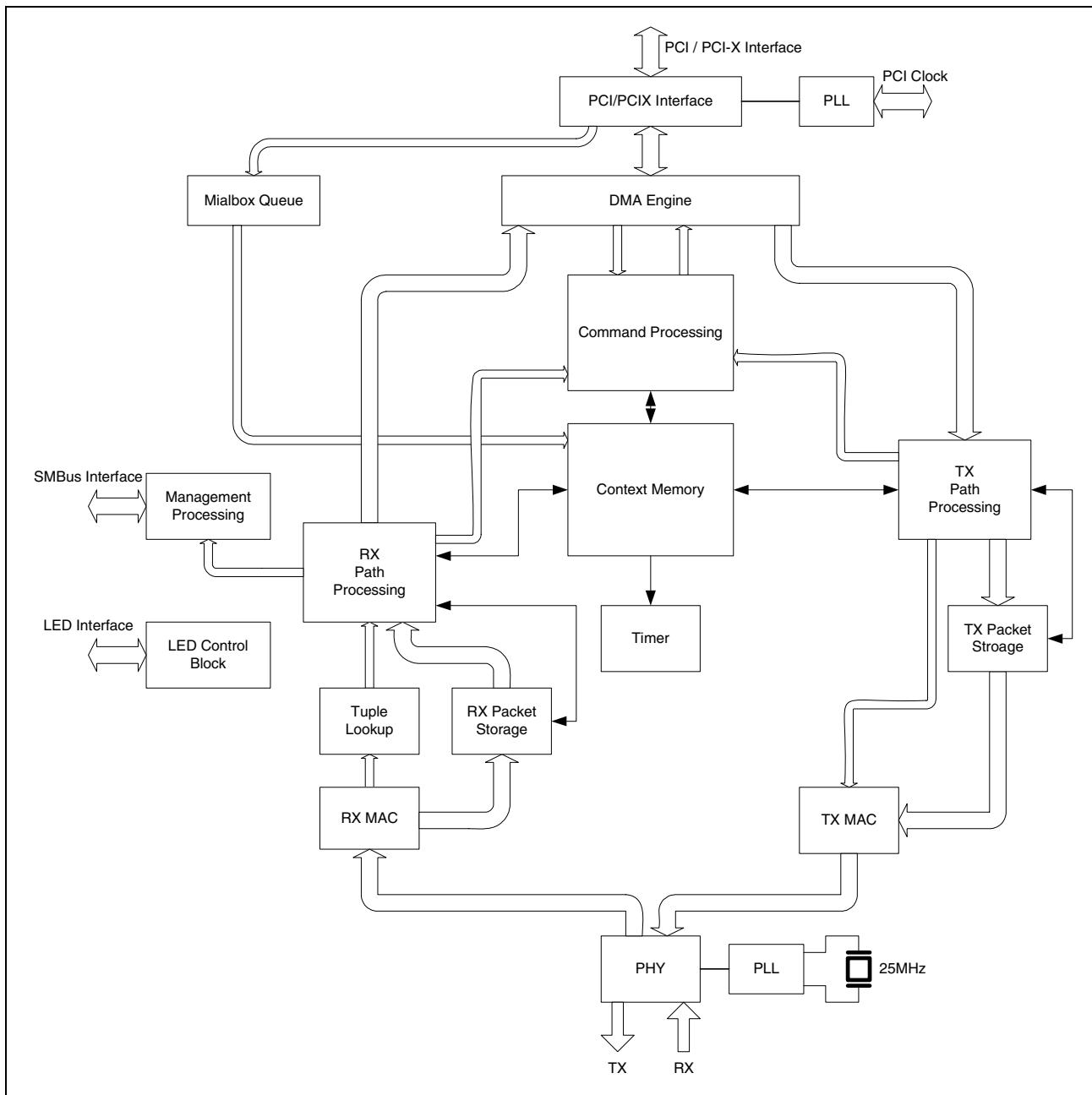


Figure 1: Broadcom BCM5706 Block Diagram

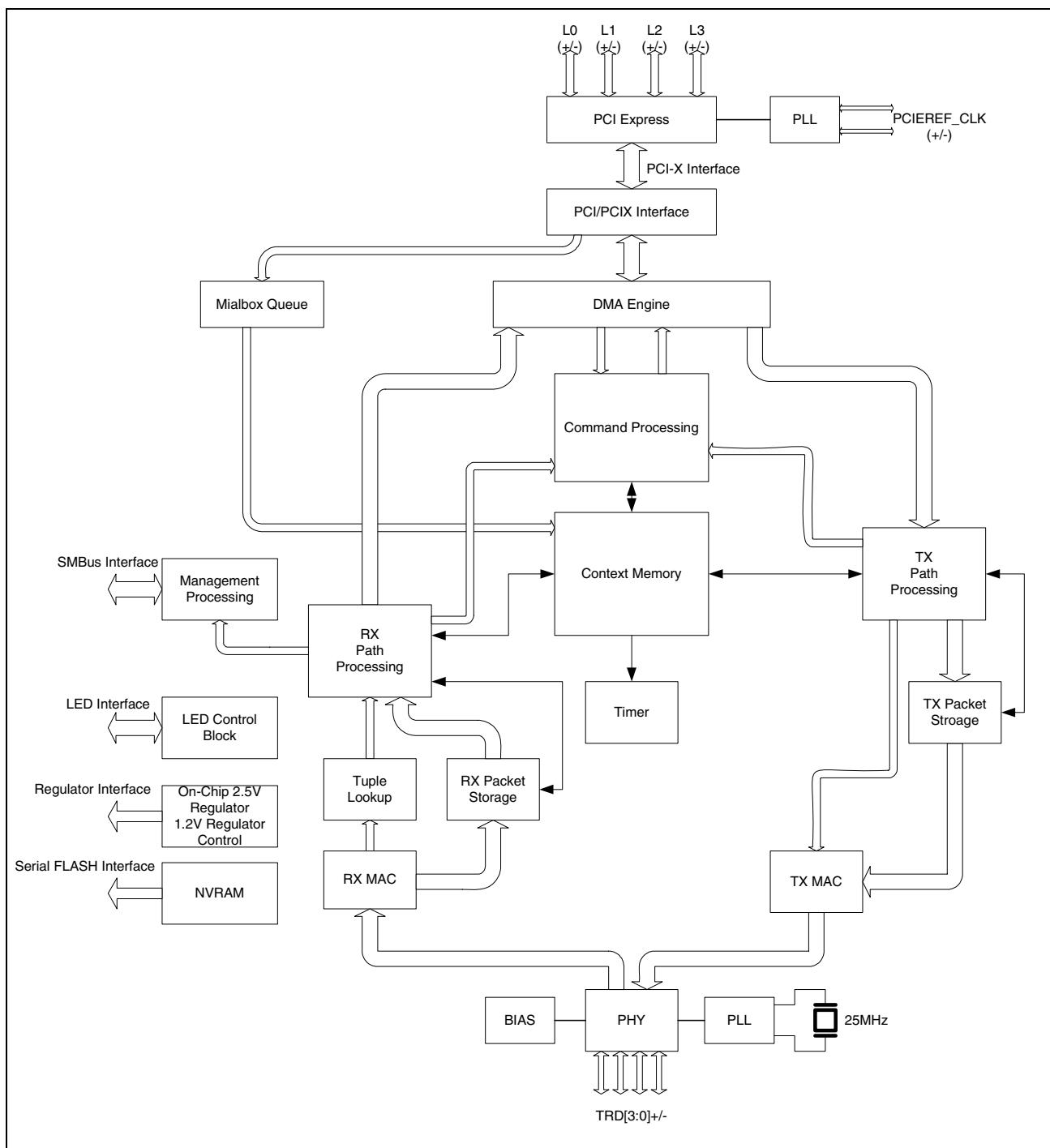


Figure 2: Broadcom BCM5708 Block Diagram

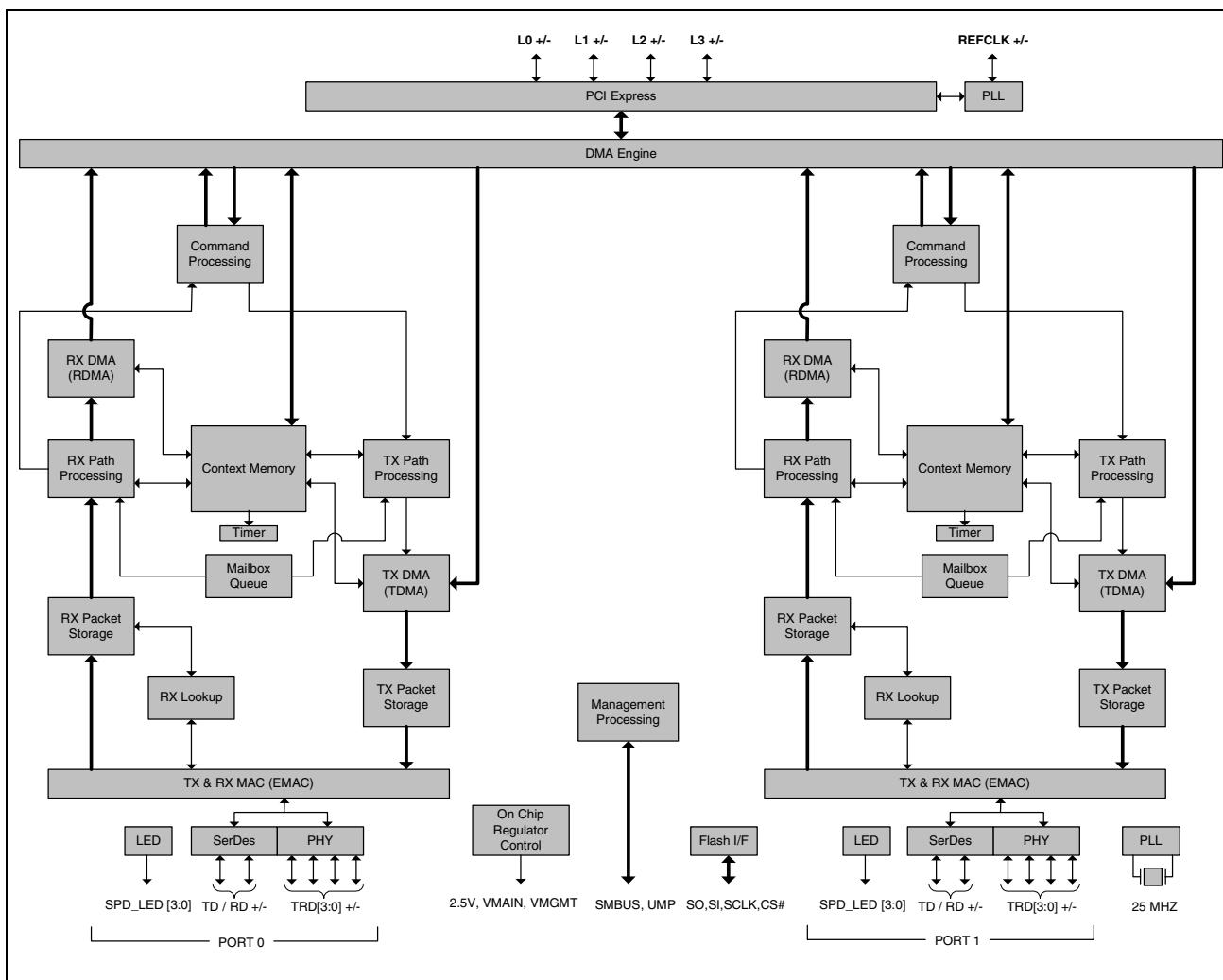


Figure 3: Broadcom BCM5709 and BCM5716 Block Diagram

OVERVIEW

The Broadcom NetXtreme II is a revolutionary Gigabit Ethernet controller that provides unprecedented functionality integrated in a network controller: full featured 10/100/1000BASE-T Gigabit Ethernet controller, TCP-offload engine (TOE), iSCSI controller, and RDMA NIC (RNIC). It raises the ante on gigabit controllers from a layer 2 device (see ISO/OSI 7 layer model comparison in [Figure 4](#)) to a layer 4 and layer 5 device integrated with a standard Gigabit Ethernet controller. It is the first Convergence NIC (CNIC) product paving the road for convergence of networking, storage, and clustering on an Internet Protocol (IP) infrastructure.

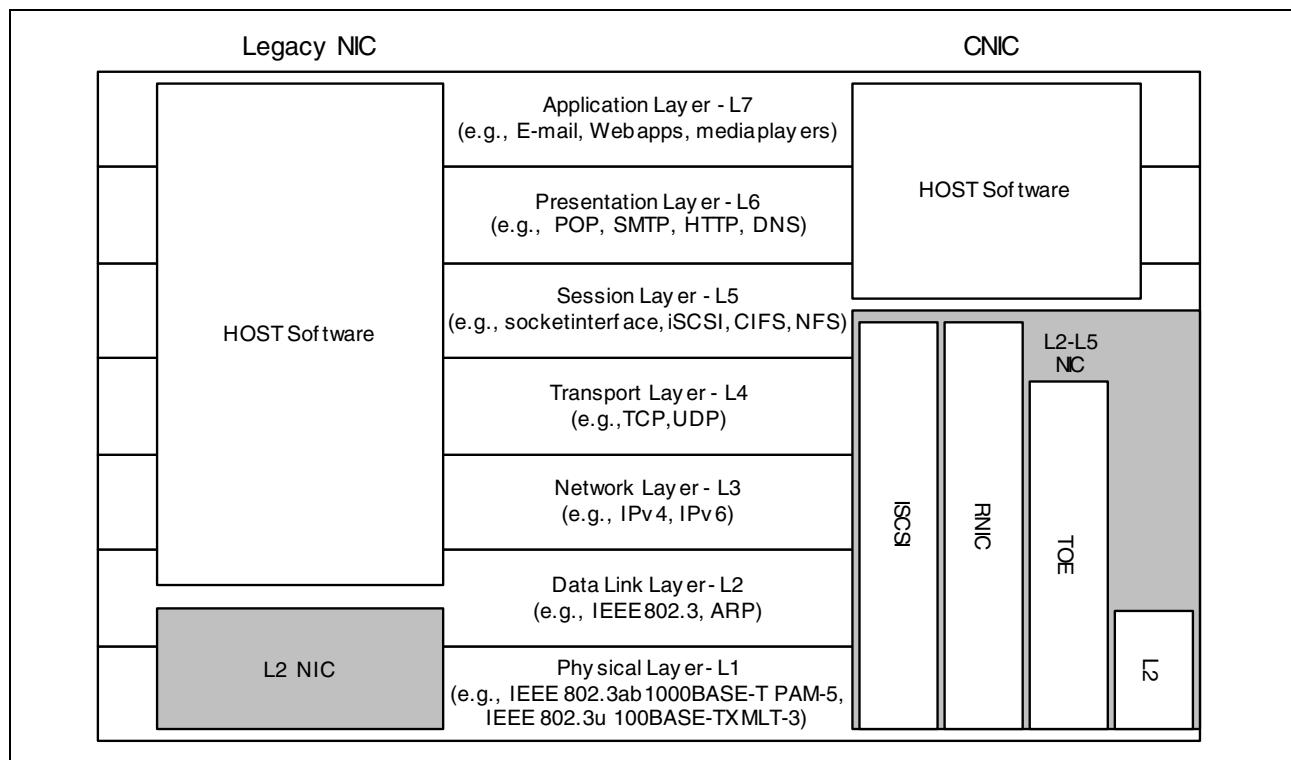


Figure 4: Layer Comparison Between Standard NIC and a TCP/IP Offload NIC

Use of RNIC (RDMA NIC) promises ultimate data transfer acceleration while remaining application or upper-level protocol (ULP) agnostic. It allows generic hardware like the Broadcom NetXtreme II to provide the best data transfer services to many different protocols with no need for a protocol-specific hardware component. The Broadcom NetXtreme II combines a network controller, storage controller, and RNIC to fully address the communications needs of servers. It allows true convergence, where all traffic is on a single network, or use of similar networks dedicated per usage. As the underlying networking technology, TCP/IP allows for reduction in cost of ownership for all of the various use models.

The highly integrated and extremely flexible architecture of the Broadcom NetXtreme II allows the user to control the hardware resource allocation per application. The user can determine how many connections are used for TOE, iSCSI, iSER, or remote DMA (RDMA) in parallel to a standard full-featured Gigabit Ethernet controller.

The Broadcom NetXtreme II includes dedicated hardware and processors that process the frames that traverse it and provide its functionality. On the transmit path, the Broadcom NetXtreme II copies the data directly from the highest hierarchy of buffers available (in prioritized order of application buffers, ULP buffers, TCP buffers) on the host; executes, when relevant, the L5 protocols; adds the iSCSI 1.0, iSER, or RDMA headers, followed by executing the TCP/IP and adding its headers, relieving the host CPU from these time-consuming operations. On the receive path, the Broadcom NetXtreme II processes the frame up to the highest layer supported present in the frame (i.e., it processes the frame for RDMA if this was an RDMA frame). Further, it removes the lower level headers from the frames it receives off the wire. It posts the data directly to application buffers, sparing host CPU resources that would otherwise be required for this activity.

For iSCSI, the Broadcom NetXtreme II provides hardware hooks for the most time-consuming tasks. On transmit, the Broadcom NetXtreme II copies data directly from the iSCSI buffers, adds framing (FIM) when used, and computes the header and data CRC when used. On receive, the Broadcom NetXtreme II strips off the framing headers and checks CRC prior to storing the data in the designated iSCSI buffers.

For RDMA, the Broadcom NetXtreme II provides direct data movement from and into the user space and kernel space (e.g., for iSER). The Broadcom NetXtreme II is RDMA Consortium (RDMAC) 1.0 compliant. On transmit, the Broadcom NetXtreme II adds the RDMA headers and CRC. On receive, it processes the RDMA headers, checks the CRC, and strips off the headers prior to storing the data in the designated ULP buffers.

TCP-OFFLOAD ENGINE

The backbone of the Internet and of the LAN, the TCP/IP protocol suite is used to provide transport (L4) services for a wide range of applications. File transfer protocols like CIFS and NFS, to name a few, utilize the services of TCP/IP. Traditionally this protocol suite was run on the host CPU consuming a very high percentage of its resources and leaving little resources for the applications themselves. Since TCP/IP has reached unprecedented ubiquity and maturity, it can be ported to hardware. In the last few years, several partial offloads have been implemented.

Example: Checksum and Large Send Offload (or TCP segmentation) have provided some relief to the host CPU utilization.

Moving the core of the TCP/IP processing to hardware is a new process: it relieves the host CPU by a ratio of up to 10x for the same amount of TCP/IP traffic processed. The Broadcom NetXtreme II provides an industry-first TCP offload engine that is carefully architected for integration with the operating system, unlike many of the standalone TOEs. In basic terms, the Broadcom NetXtreme II architecture allows the operating system to provide control and management functions (e.g., connection setup, prevention of denial-of-service attack, system resource allocation, selection of the most promising TCP/IP flows for offload, error and exception handling). The Broadcom NetXtreme II completely owns the data processing of TCP/IP flows offloaded to it. The Broadcom NetXtreme II is fully compliant with relevant Internet Engineering Task Force (IETF) RFCs and fully supports modern TCP options like time stamp, window scaling, selective acknowledgement (SACK), and so on. The implementation also provides flexibility and robustness against potential changes to TCP algorithms like congestion control by implementing this functionality in firmware, making changes easy to manage.

On transmit, the DMA engine fetches data from the host memory, segments it to the size allowed by the network (MSS or maximum segment size), formats the TCP, IP, 802.2 (e.g., VLAN), and Ethernet headers and sends it on the wire. The TCP/IP context that resides on-chip is updated accordingly (all the TCP/IP state variables and timers). If the frame has not reached its destination, the Broadcom NetXtreme II retransmits it according to TCP protocol rules. On receive, every frame is parsed, and if it is a valid frame (i.e., it passes all checks for Ethernet, IP, and TCP frame format and consistency) and it belongs to one of the connections offloaded to the Broadcom NetXtreme II, it is processed for TCP/IP. The processing includes the complete TCP/IP protocol state variables, header removal for Ethernet, IP and TCP. The data is posted in a temporary buffer similar to a software stack today or is copied directly to the buffer pre-posted by the application (i.e., Zero Copy). The Broadcom NetXtreme II fully handles out-of-order reception, including placement of data in the buffers (Zero Copy).

The Broadcom NetXtreme II TOE functionality allows simultaneous operation of up to 1024 fully offloaded TCP connections. The Broadcom NetXtreme II TOE significantly reduces the host CPU utilization while preserving the rich and flexible nature of the software implementation of the operating system stack. As the TCP control loop is shorter (ACK messages and other control functions handled by hardware), TCP data exchange becomes faster and more efficient and latencies of TCP operation are reduced.

[Figure 5 on page 21](#) shows an example of the basic flow of application data through the networking stack on its way to the wire. This data flow begins with the application writing a block of data (e.g., 64 KB) to the sockets interface. The application may insert application layer headers in the data (e.g., every 8192 bytes) and split it into 8 messages or PDUs. The networking stack further splits this into TCP segments, typically 1500 bytes each (also the typical MTU setting for Ethernet frames) to fit into standard Ethernet frames. The TCP stack then processes TCP and IP and adds the TCP/IP headers into each packet. The packet information is then posted to the NIC's output queue, and the NIC fetches the data from the host memory, calculates checksum and Ethernet CRC, and places it into the NIC buffers for transmission. The NIC transmits the information to the wire and interrupts the CPU to signal completion of the transaction.



Figure 5 also illustrates the differences between the existing technologies of checksum offload and TCP segmentation, and between TOE and RDMA. With checksum offload, the NIC calculates and appends the TCP/IP checksum to each outgoing packet, offloading the CPU of this task. TCP segmentation copies the packet header from one packet to the next in a string of like packets, further offloading the CPU of the creation of each packet header when sending large segments of data. TOE even further offloads the CPU by allowing the NIC to parse segments into packets and handle the tasks associated with reliable transport (windowing, sequence numbers, packet acknowledgements, and so on) and by completely processing the received segments, placing the data in the buffers instead of the host stack. RDMA provides the ultimate offload by allowing the RNIC to communicate directly with the application by bypassing the kernel, taking application data, creating segments, and handling TCP/IP and lower-layer functions to move the data directly into the receiver's application space.

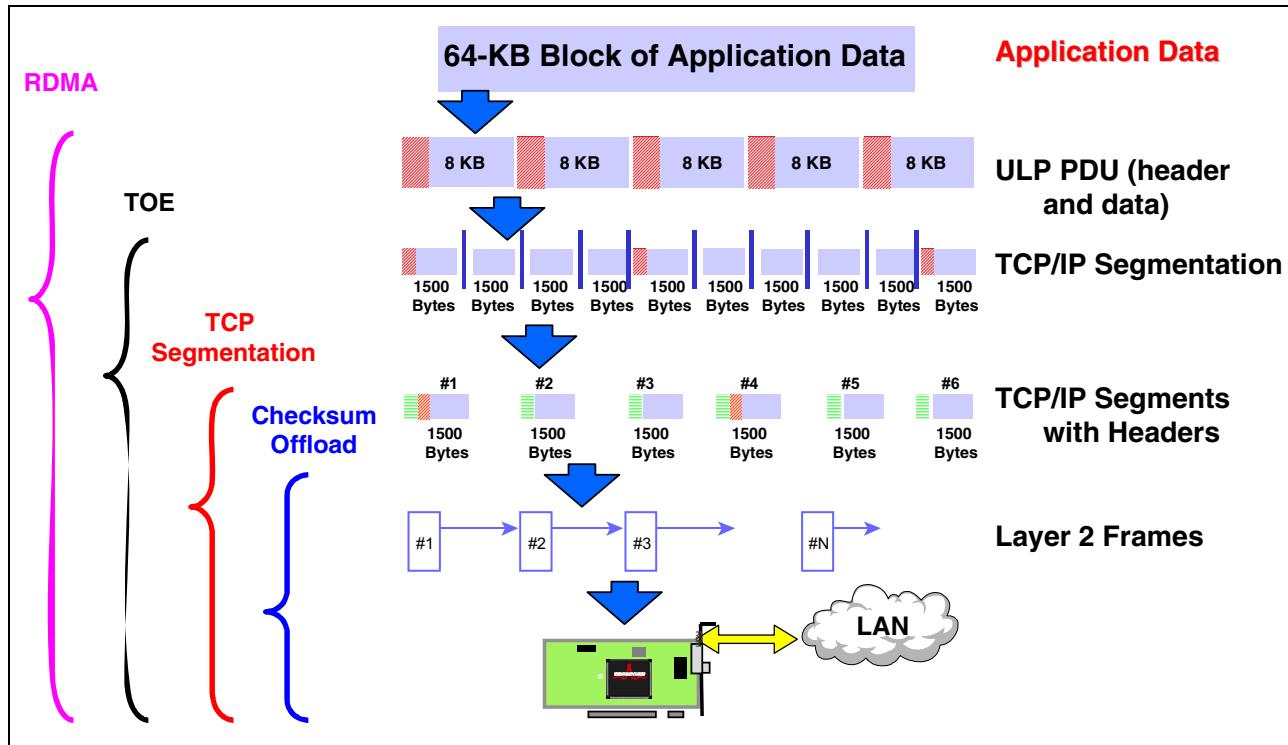


Figure 5: Data Flow From Application Through Wire

REMOTE DMA

Remote DMA (RDMA) is a technology used to allow an application or a kernel program to communicate directly with a remote peer by moving data to/from local memory to a remote memory. This technology has been pioneered by Virtual Interface Architecture (VIA), and with Infiniband® (IBTA). The Broadcom NetXtreme II supports the mapping of RDMA technology over TCP/IP as proposed by the RDMAC and is being further standardized by the IETF. Direct transfer of data by the RNIC, with no need for host CPU involvement in the data transfer (also known as kernel bypass), significantly reduces CPU utilization of data communication and drastically reduces latencies for moving data from one machine to the other. Reduced latency to the range of 10 ms–20 ms enables a broad range of high-end latency-sensitive applications to best perform over Ethernet, which is unprecedented. Applications like database, high-performance computing (HPC), and storage can now use an RNIC for very high performance.

The RDMAC has standardized a full set of wire protocols MPA/DDP/RDMA (i.e., marker PDU-aligned framing for TCP specification/direct data placement/remote DMA protocol) for mapping RDMA over TCP as well as verbs specification for defining the RNIC-to-host software interface.

With traditional DMA, the CPU first programs the DMA engine with the source address, the destination address, and length of the address block. Independently of the CPU, the DMA engine then copies the source buffer to the destination buffer and notifies the CPU of the task completion.

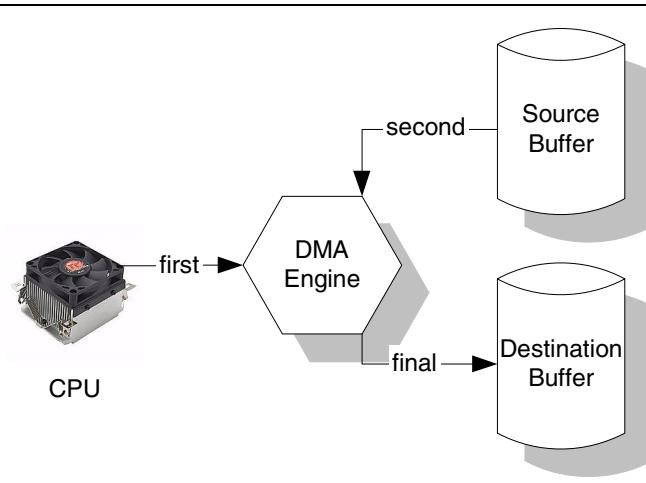


Figure 6: Traditional DMA

With RDMA, the traditional concept of DMA is extended over the network. In the first step, the two peers exchange data describing the transaction that will happen and then inform the local RNICs (if necessary). In the second step, the sending CPU programs the local RDMA engine (i.e., RNIC) with the source address, the destination host address, and the data length. The sending RNIC then fetches the data from the source buffer and transmits the data over the existing Ethernet infrastructure to the remote system. In the final step, the destination RDMA engine reconstitutes the data and places the received data into the destination buffer. Both source and destination CPU can be notified when the transaction is complete (depending on transaction specifics).

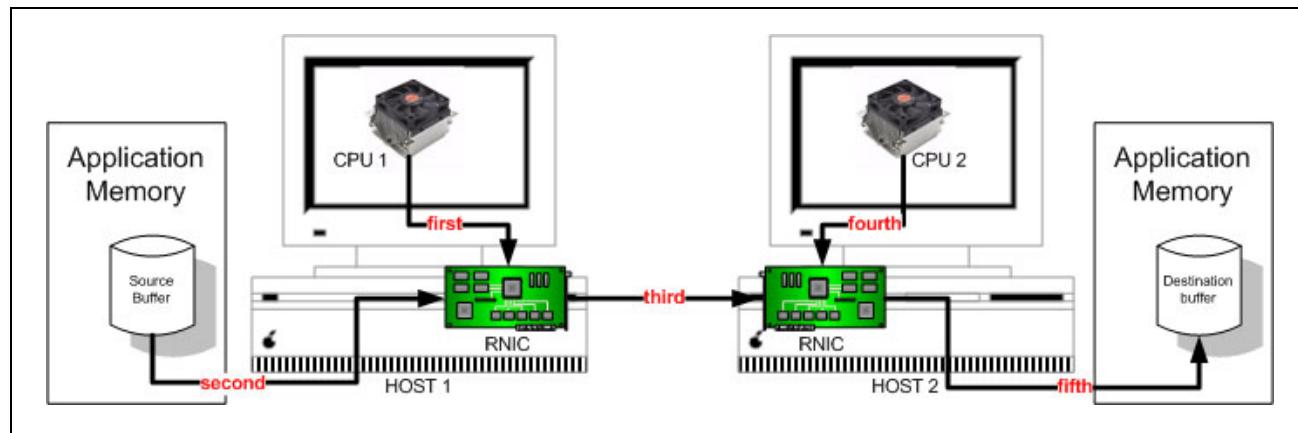


Figure 7: RDMA

The MPA protocol (marker PDU-aligned framing for TCP specification) provides alignment of the message to the TCP header and digests coverage for the whole segment (header and data) and markers to allow for direct data placement for out-of-order TCP segments. Direct Data placement protocol/remote DMA protocol (DDP/RDMA), provides the data transfer mechanisms and the control infrastructure to support it. DDP/RDMA provides for RDMA Read and Write operations and for Send operations. RDMA Read and RDMA Write allow direct data transfer to/from an application buffer by the RNIC with no software involvement. This model works best for minimal delay, for true 0-copy on the receiver all the way to the application (above the TCP level). The send operation allows for moving data into an intermediate buffer (anonymous buffer) for control and unsolicited small messages.

The Broadcom NetXtreme II fully implements the wire protocols and supports the Verbs. Its flow-through design allows direct data placement of in-order or out-of-order TCP segments with RDMA messages. No external buffer memory is required to accomplish this, reducing the cost, power, real estate, and complexity of the solution as well as the latencies.

The RDMAC has also defined two application level protocols for RDMA—iSER and SDP. iSCSI extensions for RDMA (iSER) maps the iSCSI protocol on top of RDMA using a standard RNIC. The mapping was created in a way that allows iSER to take advantage of the iSCSI 1.0 ecosystem for management (MIB), naming and discovery, logon and negotiation, bootstrapping, negotiation, and security. iSCSI can employ RDMA to get high performance for storage applications.

The sockets-direct protocol (SDP) is designed to allow standard sockets application to take advantage of RDMA infrastructure. Using SDP and its pre-standard Winsock Direct Protocol (WSD) exposes the benefits of RDMA to a broad range of applications, such as databases.

iSCSI

The IETF has standardized the Internet Small-Computer Systems Interface (SCSI or iSCSI). The SCSI is a popular family of protocols that enable systems to communicate with storage devices, using block-level transfer (i.e., address data stored on a storage device that is not a whole file, unlike file-transfer protocols such as NFS or CIFS). iSCSI maps the SCSI request/response application protocols and its standardized command set over TCP/IP networks. For further information on iSCSI, refer to [\(iSCSI\)](#).

As iSCSI utilizes TCP as its sole transport protocol, it greatly benefits from hardware acceleration of the TCP processing (i.e., use of a TOE). However, iSCSI as a layer 5 protocol has additional mechanisms beyond the TCP layer. [Figure 8](#) shows the relationship iSCSI has to TCP and to the SCSI layer. As is custom in the SCSI family, the initiator requests certain operations (e.g., IO read or IO write) from the target in an end-to-end SCSI session using iSCSI.

The iSCSI layer adds several mechanisms above and beyond the TCP transport service. The iSCSI frame shown in [Figure 9 on page 25](#) is an iSCSI PDU embedded as the payload of a standard TCP/IP frame. To further enhance the TCP checksum mechanism, iSCSI optionally uses header and data digest (i.e., CRC-32c). To assist the receiver in data placement, iSCSI has devised a dedicated (though optional) sync and steering layer, utilizing a fixed interval marker (FIM). As iSCSI exposes the use of a named buffer to both the initiator and the target, it also facilitates for direct data placement into these named buffers. These mechanisms benefit from hardware acceleration.

The Broadcom NetXtreme II targets best-system performance, maintains system flexibility to changes, and supports current and future OS convergence and integration. Therefore, the Broadcom NetXtreme II iSCSI offload architecture is unique as evident by the split between hardware and host processing. Unlike a monolithic implementation on an HBA, the Broadcom NetXtreme II focuses mainly on offloading to hardware the time-consuming elements poorly handled by software. A monolithic implementation offloads the complete iSCSI protocol to the HBA. This adds complexity to the HBA in order to process the complete control plane, which is better handled by the host CPU, and limits the integration options with a host operating system (e.g., control plane on the host provides for flexible implementation and robustness against updates to the iSCSI protocol, allowing for virtually no limit on number of outstanding commands and no limit on number of connections per session, etc.). The Broadcom NetXtreme II accelerates in hardware the iSCSI mechanisms that impact system resource utilization and performance. It supports all of these mechanisms with the support of specially built hardware circuitry or firmware executing on-chip. The Broadcom NetXtreme II provides rich and complete iSCSI HBA functionality.

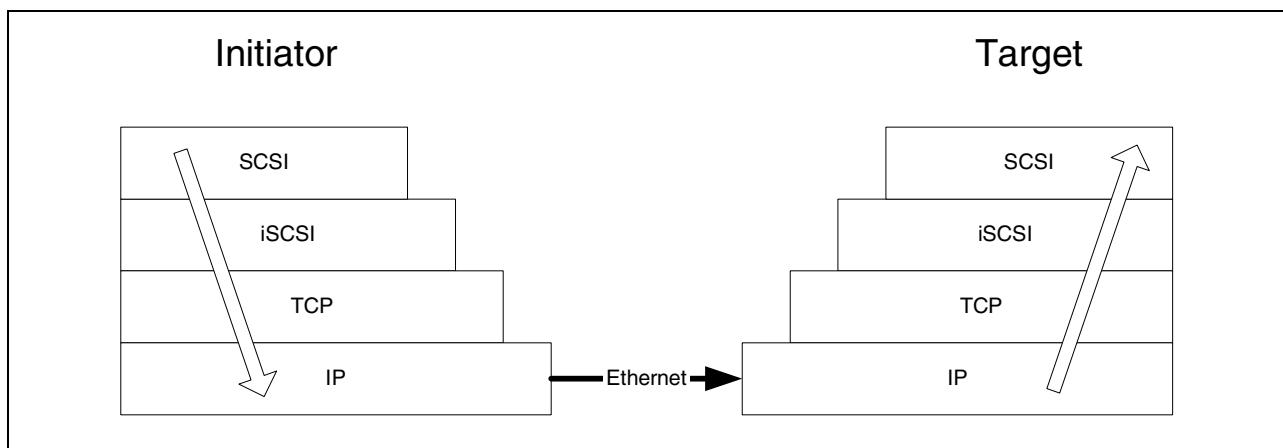


Figure 8: iSCSI Layers

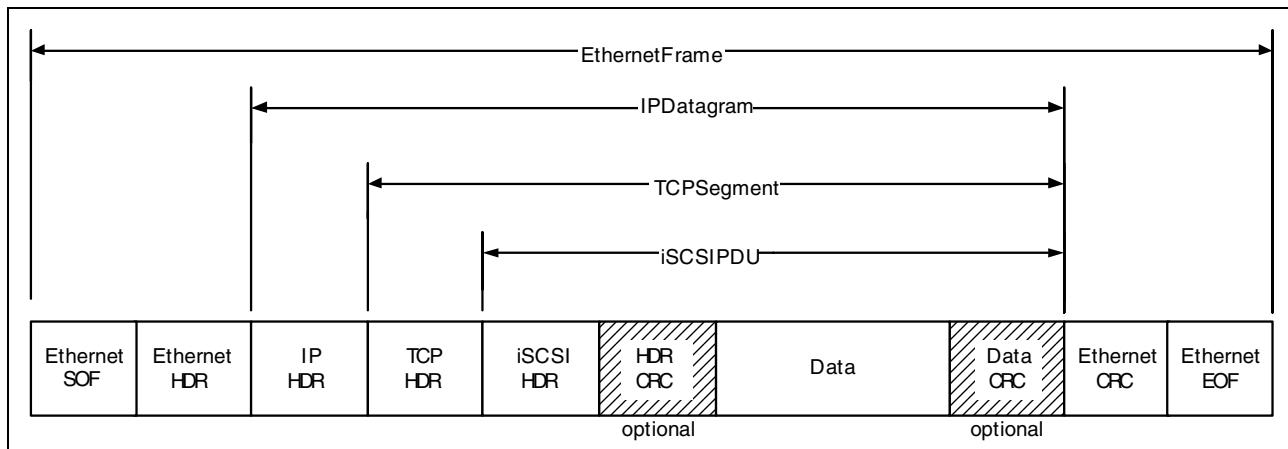


Figure 9: iSCSI Ethernet Frame Encapsulation

The Broadcom NetXtreme II supports the iSCSI frame format, header/data separation, insertion and checking of the iSCSI header and data CRCs, insertion and deletion of the fixed interval marker (FIM), and direct data placement into the named iSCSI buffers. A deep command queue ensures optimal utilization of the network for maximal bandwidth and a high number of I/O operations per second (IOPS).

[Figure 10](#) depicts a typical data flow between an iSCSI initiator and an iSCSI target. The Broadcom NetXtreme II can be the core of an initiator or target HBA. The initiator's SCSI driver passes SCSI control data blocks (CDBs) to the iSCSI layer that sits atop the TCP/IP stack. On the target, a storage device (i.e., SCSI disk) sees SCSI commands and has no knowledge of the iSCSI layer. The SCSI disk interprets the SCSI commands, executes them, and sends appropriate data or replies to its local iSCSI layer, which transports them to the iSCSI on the initiator.

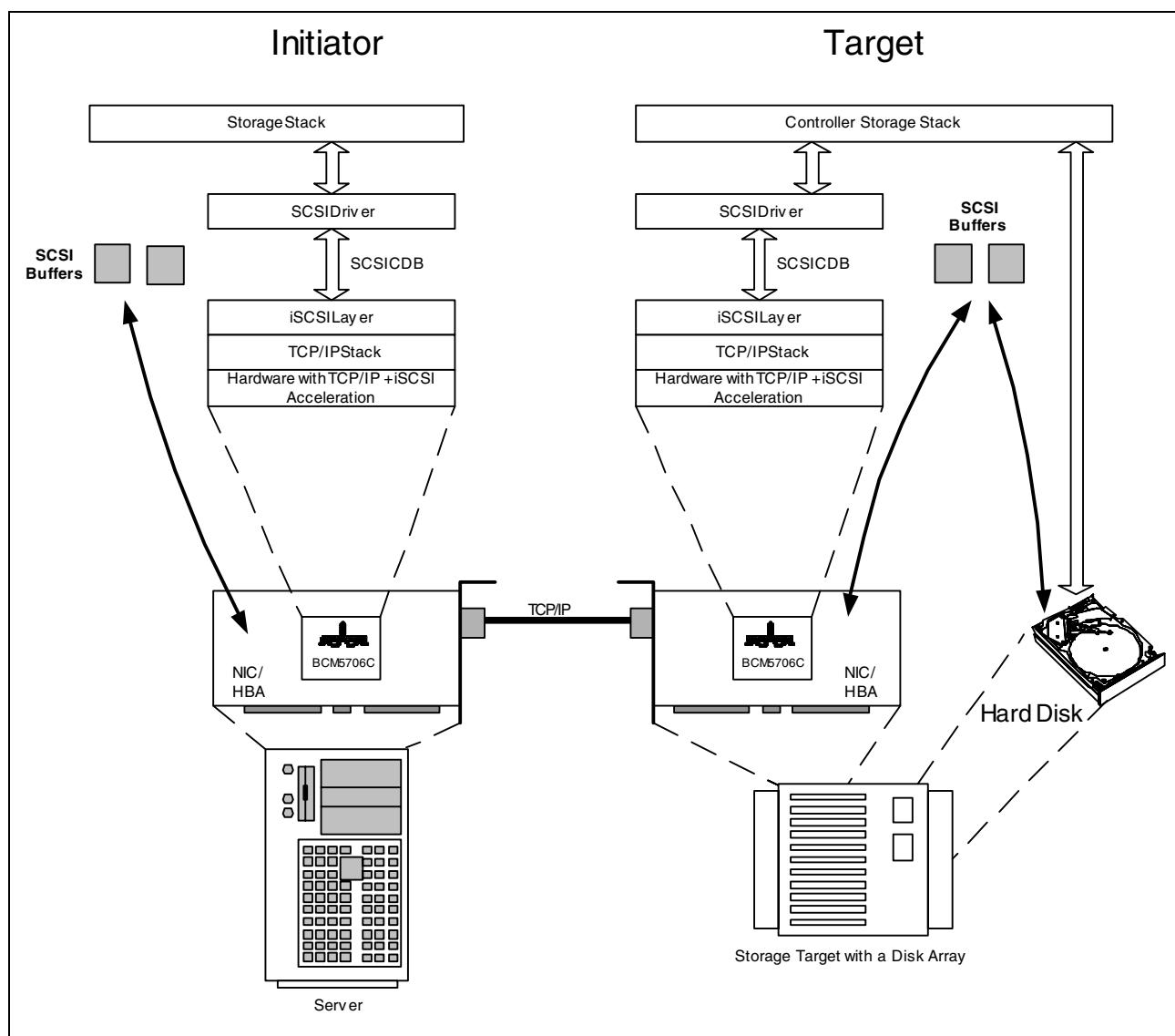


Figure 10: iSCSI Architecture

HOST MEMORY DATA STRUCTURES

All host memory structures that pass data between the Broadcom NetXtreme II and driver must be in locked system memory. Locking prevents memory from being swapped to disk or being moved so that it maintains a consistent physical memory address from the device's point of view.

Data buffers are used to store payload data that is moving between the Broadcom NetXtreme II, driver, host operating system, and application. These are always allocated by the operating system. On host operating systems that provides virtual memory, the allocated memory will appear to exist at different addresses (and may even appear to be fragmented) depending on the point of view.

In [Figure 11](#), the driver sees a single, contiguous virtual memory block, while the Broadcom NetXtreme II sees the same memory as four discrete physical memory blocks because the host operating system will allocate multiple pages (e.g., 4 KB on many systems). For the data buffer to be accessible by the Broadcom NetXtreme II hardware, it must be locked into host physical memory and converted into a list of one or more physical address/length pairs. Normally, this physical view of a data buffer is carried in a Buffer Descriptor Chain as described in [Figure 11](#).

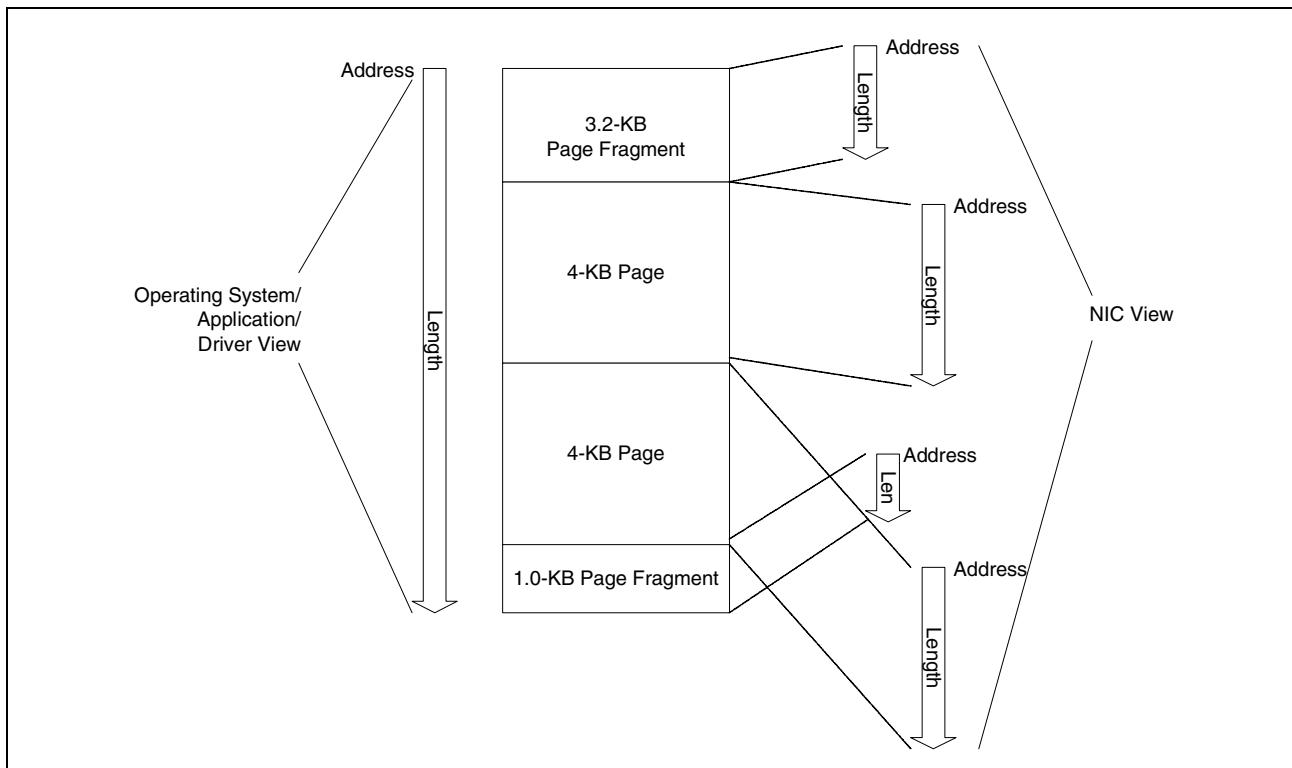


Figure 11: Virtual Address vs. Physical Address Views

Because the Broadcom NetXtreme II supports bus-mastering, it is able to update chains independently of the driver. To avoid a complicated mechanism for locking access to records that are shared by the Broadcom NetXtreme II and the driver, chains will be used in one of two directions. In the case of packet data being sent out onto the Ethernet, the driver is typically adding data to a chain and the Broadcom NetXtreme II is typically removing data from a chain. In this case, the driver is referred to as a *writer* and the Broadcom NetXtreme II is referred to as a *reader*. In the receive direction, these roles are reversed and the Broadcom NetXtreme II becomes the *writer* while the driver takes on the role of a *reader*.

Because a reader and writer may access a chain independently of each other, the writer will maintain a producer index (also known as the head) which points to the next entry to be written, while the reader will maintain a consumer index (also known as the tail), which will track the next entry to be read. When the producer and consumer index are the same, the chain is empty.

At reset, all indexes are 0, and they increment by one for every record passed in the chain. If records are skipped at the end of a chain page, the index will be incremented to account for this. The record offset within a chain page can always be determined from the index (index MOD recs_per_page).

BUFFER DESCRIPTOR CHAIN

Chains hold a variable number of records of identical size. The purpose of the chain structure is to allow the driver and Broadcom NetXtreme II to communicate through a pre-allocated host memory area without having to work with physical addresses. A chain is made up of one or more pages of memory. The system allocation page size (usually 4 KB) is a useful size for a chain page.

The driver must first allocate a large number of pages, lock them into physical memory, and create a free list of these pages called a Free Page Chain (see [Figure 12 on page 29](#)). Several of these Free Page Chains may be created.

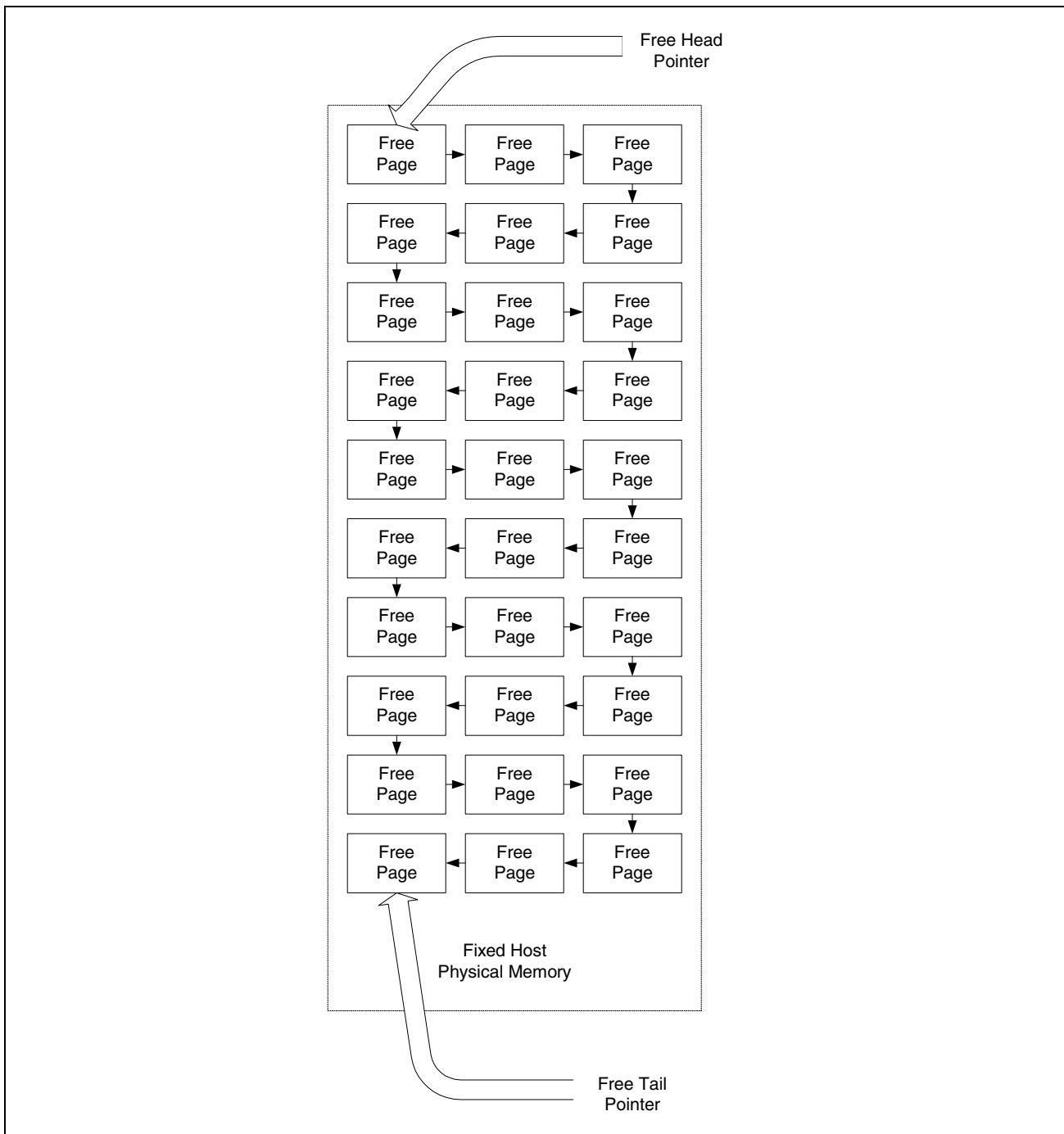


Figure 12: Free Page Chain

Each page will hold an integer number of fixed size records (usually 16 bytes). The writer of a chain must then allocate a free page from the Free Page Chain before it can write a record entry within a page. If the Broadcom NetXtreme II is writing the chain, then the Free Page Chain must be passed to the chip by the driver. When the number of records exceeds what will fit in a single page, the writer must allocate additional pages from the Free Page Chain and link them to the active chain using a special link record (see [Figure 13](#)).

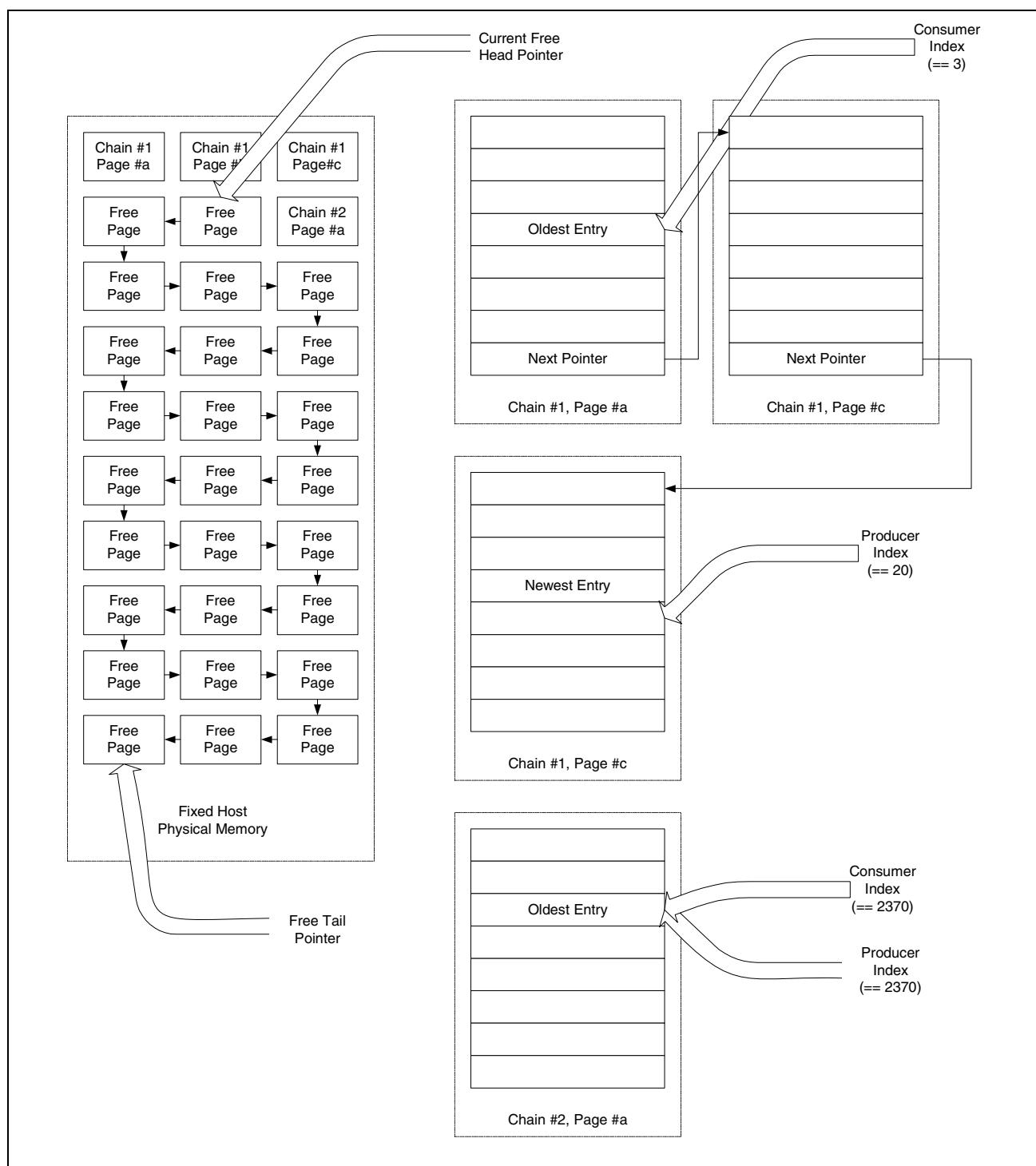


Figure 13: Chain Format with Multiple Pages

The record format for each chain type is defined in “Common Data Structures” on page 73.

NORMAL RECORD ENTRY

A Normal Record Entry consists of a single record. The flags field of the Entry identifies it as a Normal Record Entry to the reader. The producer/consumer index is incremented for every Normal Record Entry.

MULTI-RECORD ENTRY

Most records fill only one entry, but some chains require larger amounts of data to be passed occasionally. In these cases, a Multi-Record Entry will be defined. The flags field of the Entry identifies it as a Multi-Record Entry to the reader. The producer/consumer index is incremented for every entry record of a Multi-Record Entry. Multi-Record entries are not allowed to cross the end of a page.

NEXT POINTER RECORD ENTRY

Each chain type will define a Next Pointer record that has the first 32 bits as a flag, 32 bits of reserved area, and a 64-bit host physical address of the next Chain page. When the next record (Normal or Multi) will use up or exceed the current Chain page, a new page must be allocated. The last record of the current page will be written as a pointer to the new page and the record entry will be written as the first record in the new page. The writer of the chain must insure that a valid next pointer is written whenever the last entry of a page is used so that the reader can read the next pointer with the last piece of data. This prevents a short, wasteful DMA at the start of the next read of records.

Next Pointers are not limited to the last record of a page. They can appear anywhere in the page list. This is important for the handling of Multi-Record entries to make sure that they do not span chain pages.



Note: RX BD chains always have 16-byte entries and the Next Pointer records are modified in two ways:

- The first 64-bits is the back pointer, pointing to the previous chain page instead of a flag field.
- The Next Pointer record always occurs in the last entry of a page.

Once a chain page is consumed, it should be returned to the free pool of pages for re-use. This allows the chains to grow and shrink without locking/unlocking physical memory.

PAGE TABLES

Page tables are used to inform the NetXtreme II of the physical mapping of large virtual spaces of physical memory for many types of uses. Whenever kernel bypass is used, all memory access must be directed through a page table.

PAGE TABLE FORMAT

Page tables hold a fixed (at creation time) number of 8-byte page table entries. Each entry maps a virtual page to a physical page. The size of the page is limited to the largest physically contiguous pieces of memory that the driver can allocate. A typical minimum possible page table can address 2 MB of memory ($[4096/8]*4096 = 2\text{ MB}$). In addition to the page table, the chip must also have base/bounds checking information and the page size referenced by all entries in the table.

ADDRESS SPACE

The registers and memory on the Broadcom NetXtreme II support multiple access methods. This section discusses the different address spaces implemented on the Broadcom NetXtreme II.

HOST MEMORY MAP

The Broadcom NetXtreme II provides host access to registers and internal memory through memory mapped I/O (see [Figure 14](#)).

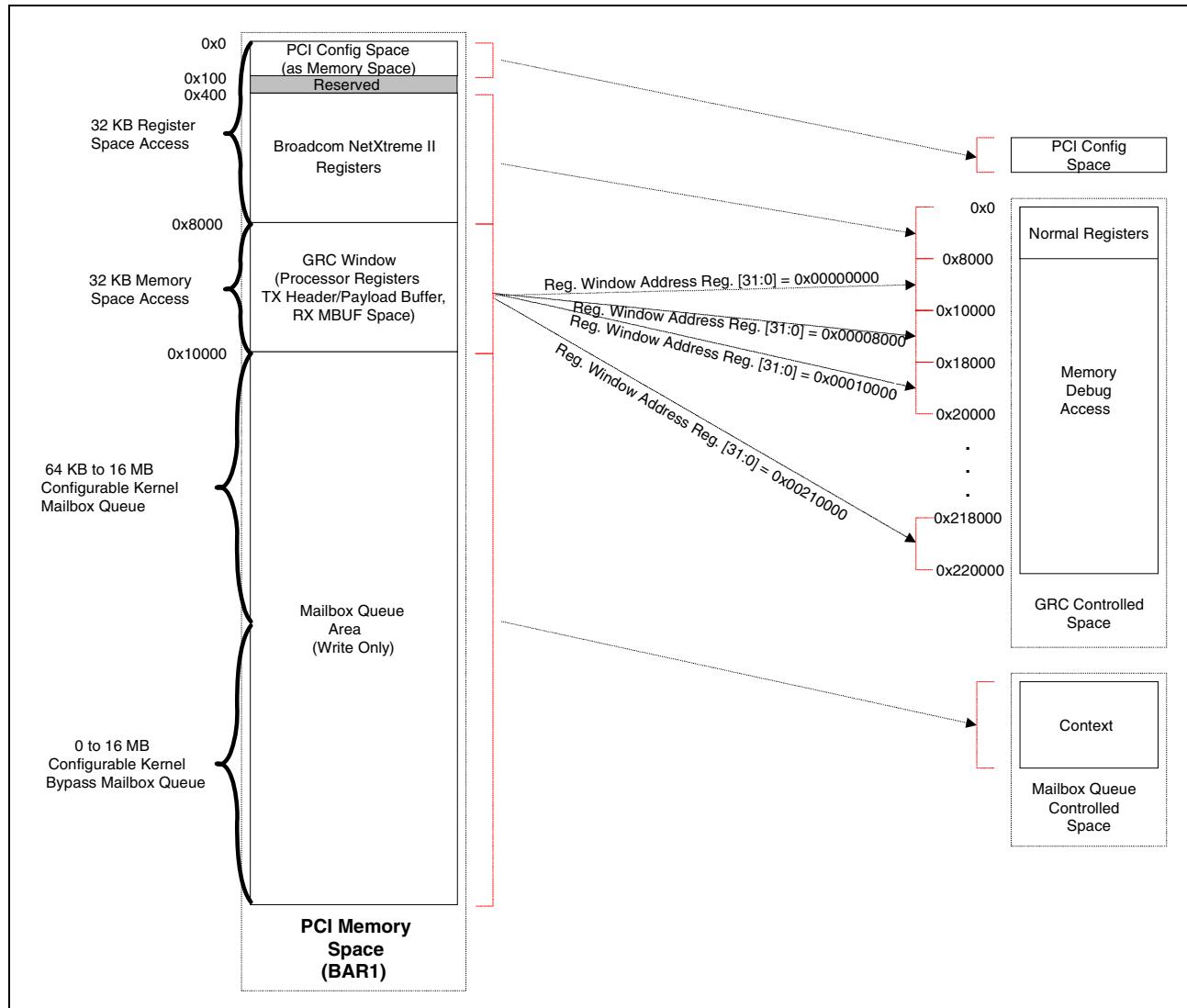


Figure 14: Host Memory Map

INTERNAL PROCESSOR ADDRESS SPACE

The Broadcom NetXtreme II contains six high-speed, 32-bit RISC processors. Each processor has a unique view of memory, though they are all roughly divided as indicated in [Figure 15](#).

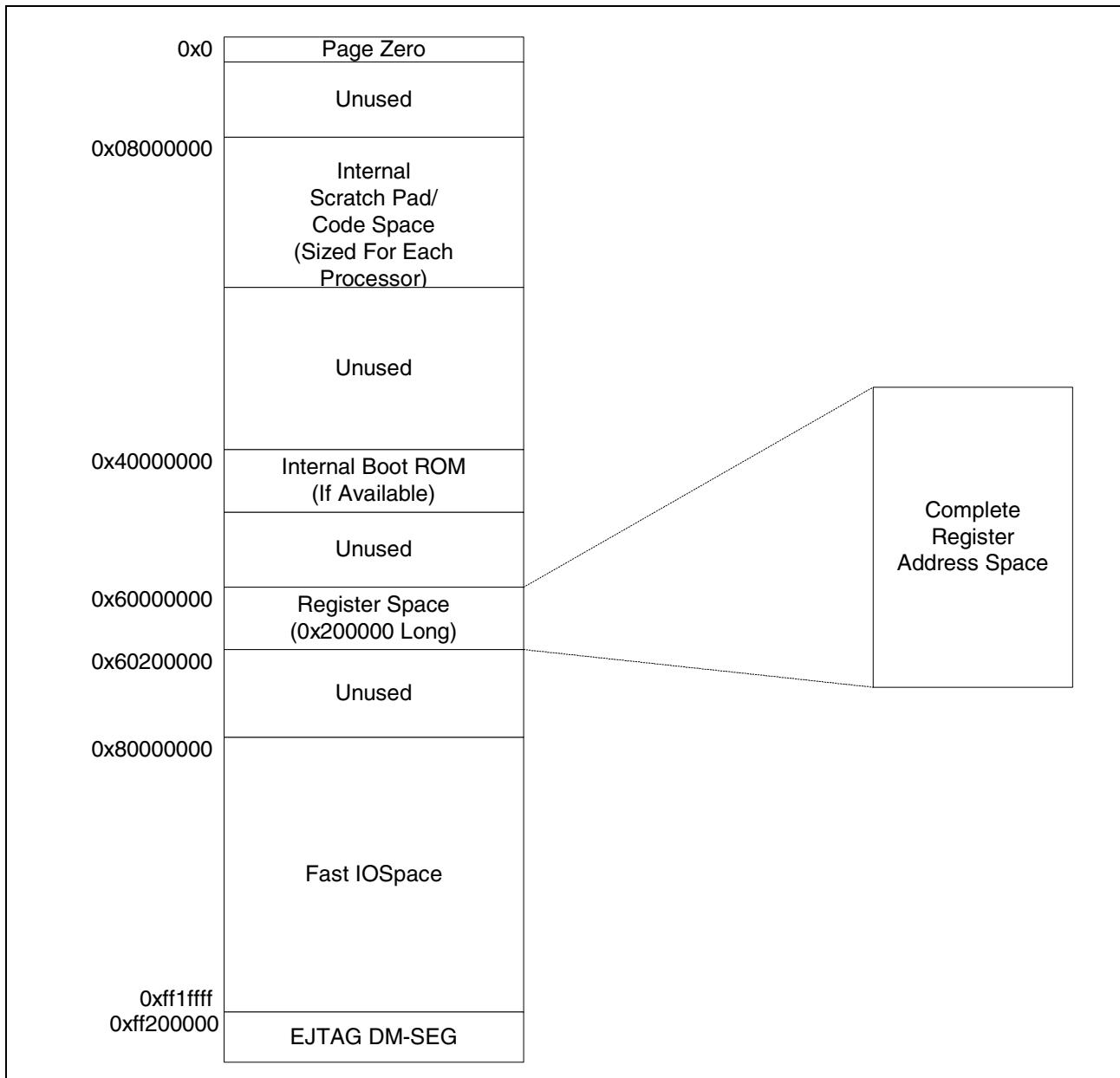


Figure 15: Internal Processor Address Space

[Table 4 on page 34](#) describes how the scratchpad memory for each processor is sized.

Table 4: Internal Processor Scratchpad Memory Sizes

Processor	BCM5706 (Size in KB)	BCM5708 (Size in KB)	BCM5709 (Size in KB)	BCM5716 (Size in KB)
TX Processor (TXP)	32	32	32	32
RX Processor (RXP)	32	36	40	40
TX Patchup Processor (TPAT)	16	12	12	12
Completion Processor (COM)	32	40	40	40
Command Processor (CP)	32	40	40	40
Management Processor (MCP)	32	32	64	64

Section 3: NVRAM Configuration

NVRAM MAP

The NVRAM contains the following information:

- Bootstrap
- Code directory
- Manufacturing information
- Feature configuration information
- VPD-R information
- Licensing information
- Programs (PXE or management firmware images)

The fields in the NVRAM are stored in big-endian format.

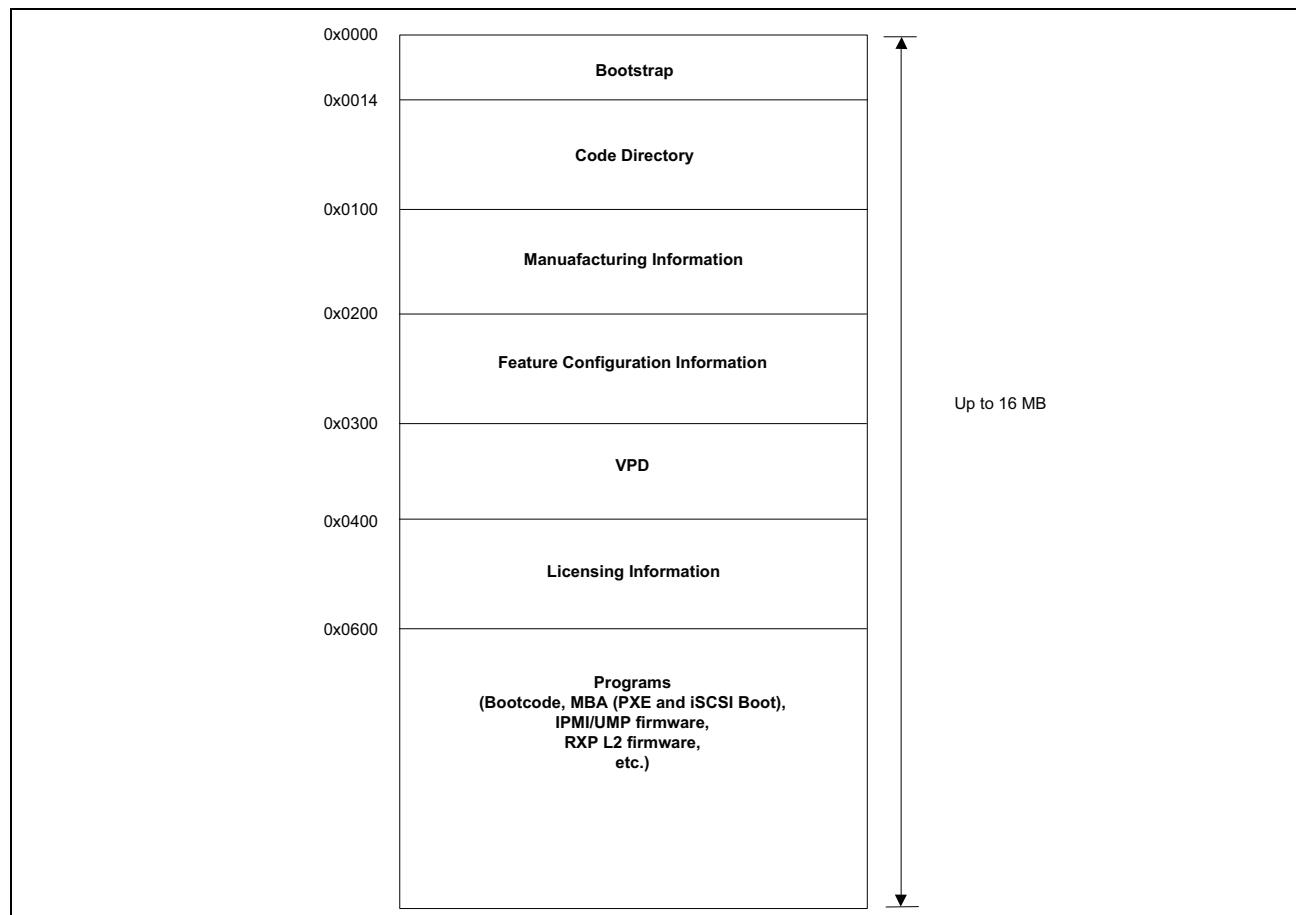


Figure 16: NVRAM Memory Map

BOOTSTRAP

The bootstrap is 20-bytes long and its format is depicted in the following table.

Table 5: Bootstrap Format

Offset	Size (Bytes)	Description
0x00	4	Magic number 0x669955aa which is used by the ROM code in Broadcom NetXtreme II controller to identify valid bootstrap.
0x04	4	CPU scratch physical address where bootstrap will be copied to and execute.
0x08	4	Bootstrap code len in DWORD. This length includes the 4-byte CRC-32 stored at the end of the bootstrap code.
0x0c	4	Offset within NVRAM where bootstrap code is found.
0x10	4	CRC-32 of the bootstrap header (offset 0x0 to 0x0f inclusive).

There is up to 1 KB of ROM code in the Broadcom NetXtreme II family that is executed by the Management Processor (MCP) when the Broadcom NetXtreme II is removed from reset. The major function of this ROM code is to load the initial bootstrap from the NVRAM to the scratch pad of the MCP.

CODE DIRECTORY

The code directory is a table of executable binaries. Each binary may be loaded onto one of the embedded RISC processors or it may be used by the host CPU. The executable images are located in the map region called *Programs and PXE Image* (see “[Program Images](#)” on page 51).

The code directory region contains 16 directories of code with 12 bytes in each directory. The CRC-32 of each binary image is appended at the end of that image to allow for integrity checking. The total size of this region is 256 bytes and its format is depicted in [Table 6](#).

Table 6: Code Directory Region

Offset	Size (Bytes)	Description
0x14	4	Directory #0 Code SRAM address, relative to the designated CPU (see attribute field at offset 0x18).
0x18	4	Directory #0 image type, attributes, and length: <ul style="list-style-type: none"> • Bit 31–28: Image Type. <ul style="list-style-type: none"> - 0x0: BC2 (second-phase bootcode) - 0x1: MBA (multiple boot agent host software) - 0x2: L2_RXP (L2 receive processor firmware, required for VAUX operation of management firmware) - 0x3: UMP (firmware) - 0x4–0x6: Reserved - 0x7: IPMI_CFG (IPMI configuration data)

Table 6: Code Directory Region (Cont.)

Offset	Size (Bytes)	Description
0x18 (cont.)	4	<ul style="list-style-type: none"> - 0x8: IPMI_INIT (IPMI initialization firmware) - 0x9: IPMI_SERV (IPMI service firmware) - 0xa: USER_BLOCK - 0xb: ISCSI_BOOT_CPRG (iSCSI boot configuration program) - 0xc: ISCSI_BOOT_CFG (iSCSI boot configuration data) - 0xd: ISCSI_BOOT (iSCSI boot host driver software) - 0xe–0xf: Reserved • Bit 27: Reserved • Bit 26–24: Target CPU <ul style="list-style-type: none"> - 0x0: None - 0x1: Host - 0x2: TXP - 0x3: TPAT - 0x4: COM - 0x5: RXP - 0x6: CP - 0x7: MCP - 0x8–0xf: Reserved • Bit 23–2: Indicates the length of the image (plus the 4-byte CRC-32) in DWORD. It implies that the maximum size per directory entry is 16 MB. • Bit 1–0: Reserved
0x1c	4	Directory #0 NVRAM Offset to image
0x20	12	Code Directory #1
0x2c	12	Code Directory #2
0x38	12	Code Directory #3
0x44	12	Code Directory #4
0x50	12	Code Directory #5
0x5c	12	Code Directory #6
0x68	12	Code Directory #7
0x74	12	Code Directory #8
0x80	12	Code Directory #9
0x8c	12	Code Directory #10
0x98	12	Code Directory #11
0xa4	12	Code Directory #12
0xb0	12	Code Directory #13
0xbc	12	Code Directory #14
0xc8	12	Code Directory #15
0xd4	20	Reserved
0xe8	20	Spare part number (similar to the PN field in the VPD), reserved for OEM use
0xfc	4	CRC-32 of the code directory (offset 0x14 to 0xfb inclusive)

MANUFACTURING INFORMATION

This region contains manufacturing information and has a total size of 256 bytes. All reserved fields should be written as 0.

Table 7: Manufacturing Information Region

Offset	Size (Bytes)	Description
0x100	1	Manufacturing Information table revision in ASCII. The current version is 'A.'
0x101	1	Reserved.
0x102	2	Length of manufacturing information (currently 0x100 for a revision 'A' manufacturing table).
0x104	2	PCI Vendor D (BCM5706 and BCM5708 only).
0x106	2	PCI Device D (BCM5706 and BCM5708 only).
0x108	2	PCI Subsystem Device D (BCM5706 and BCM5708 only).
0x10a	2	PCI Subsystem Vendor D (BCM5706 and BCM5708 only).
0x10c	16	NULL-terminated part number of the device. This field is identical to the PN field in the VPD-R region.
0x11c	1	Power dissipated in the D3 state. Note: The data scale is hard coded at 0.1.
0x11d	1	Power dissipated in the D2 state. The NetXtreme II family does not support the D2 state. This value should always be set to 0.
0x11e	1	Power dissipated in the D1 state. The NetXtreme II family does not support the D1 state. This value should always be set to 0.
0x11f	1	Power dissipated in the D0 state. Note: The data scale is hard coded at 0.1.
0x120	1	Power consumed in the D3 state. Note: The data scale is hard coded at 0.1.
0x121	1	Power consumed in the D2 state. The NetXtreme II family does not support the D2 state. This value should always be set to 0.
0x122	1	Power consumed in the D1 state. The NetXtreme II family does not support the D1 state. This value should always be set to 0.
0x123	1	Power consumed in the D0 state. Note: The data scale is hard coded at 0.1.
0x124	4	Shared hardware configuration: <ul style="list-style-type: none">• Bits 31–20: Reserved• Bit 19: SMBus timing (BCM5709 and BCM5716 only)<ul style="list-style-type: none">- 0 = 100 kHz- 1 = 400 kHz• Bit 18: PCIe Gen 2 Support (BCM5709 and BCM5716 only)<ul style="list-style-type: none">- 0 = Disabled- 1 = Enabled• Bits 17–16: Dual MAC mode (BCM5709 and BCM5716 only)<ul style="list-style-type: none">- 00b: Normal. Both MACs available.- 01b: Reserved- 10b: Function 0 enabled, function 1 hidden- 11b: Reserved

Table 7: Manufacturing Information Region (Cont.)

Offset	Size (Bytes)	Description
0x124 (cont.)	4	<ul style="list-style-type: none"> • Bits 17–16: LED Application (BCM5708S A0 and B0 only) <ul style="list-style-type: none"> - 00b: Independent LEDs (common for a NIC design) - 01b: Multi-color LEDs - 10b: All speeds tied together (common for a LOM design) • Bit 15: Allow Gigabit Link in VAUX <ul style="list-style-type: none"> - 0: Disable - 1: Enable <p>Note: This setting also requires that “VAUX Current Overdraw” (bit 2) also be enabled.</p> • Bits 14–12: Firmware load SPIO toggle (BCM5709 and BCM5716 only). If management firmware is enabled then the management firmware is loaded as follows: <ul style="list-style-type: none"> - 000b: Load any management firmware present in the following order: NC_SI, UMP, IPMI - 001b: Load only NC_SI firmware. (If no NC_SI firmware is present, then no management firmware is loaded.) - 010b: Load only UMP firmware. (If no UMP firmware is present, then no management firmware is loaded.) - 011b: Load only IPMI firmware. (If no IPMI firmware is present, then no management firmware is loaded.) - 100b: Use SPIO4 to select between NC_SI (0) and IPMI (1) - 101b: Use SPIO4 to select between UMP (0) and IPMI (1) - 110b: Use SPIO4 to select between NC_SI (0) and UMP (1) - 111b: Reserved • Bits 14–12: Firmware load GPIO toggle—If management firmware is enabled, and both IPMI and UMP firmware are present, the value of this GPIO is sampled after reset to determine which management firmware to load. If the GPIO is sampled as 0, UMP is loaded; otherwise, IPMI is loaded. <ul style="list-style-type: none"> - 000b: Disabled - 001b: Reserved - 010b: Reserved - 011b: GPIO3 - 100b: GPIO4 - 101b: GPIO5 - 110b: GPIO6 - 111b: GPIO7 • Bit 11: Enable UMP PHY Timing (BCM5708 B0 and later, BCM5709 and BCM5716)—This setting is hardware design dependent and should be enabled when the UMP’s MII/RMII interface is directly connected to a PHY. In this mode the TX_CLK and RX_CLK are driven as outputs and are synchronized to the CK25 output. This option should be disabled when connected directly to another MAC. In this mode the TX_CLK and RX_CLK pins are selected as inputs and are driven by the CK25 output. <ul style="list-style-type: none"> - 0: Disable - 1: Enable

Table 7: Manufacturing Information Region (Cont.)

Offset	Size (Bytes)	Description
0x124 (cont.)	4	<ul style="list-style-type: none"> • Bits 10–8: PHY LED mode <ul style="list-style-type: none"> - 000b: MAC mode - 001b: PHY mode 1 - 010b: PHY mode 2 - 011b: PHY mode 3 (BCM5708 only) - 100b: PHY mode 4 (BCM5708 only) - 101b: PHY mode 5 (BCM5708 only) - 110b: PHY mode 6 (BCM5708 only) - 111b: PHY mode 7 (BCM5708 only) • Bits 10–8: Reserved (BCM5709 and BCM5716 only) • Bit 7: CRS/RXDV UMP Selection (BCM5708 B0 and later) <ul style="list-style-type: none"> - 0: RXDV (UMP uses an RMII interface, the UMP_CRS pin is connected to the PHY CRS pin, and the PHY's CRS pin drives CRS/DV) - 1: CRS • Bit 6: Backplane application (BCM5708S and BCM5709S only) <ul style="list-style-type: none"> - 0: The LOM is not used in a backplane application - 1: The LOM is used in a backplane application • Bit 5: Enable 2.5G support (BCM5708S and BCM5709S only) <ul style="list-style-type: none"> - 0: Disable 2.5G support - 1: Enable 2.5G support <p>Note: 2.5G support is only enabled when both this bit is set to enable and the appropriate license key with 2.5G support is installed.</p> • Bit 4: PCIe beacon in WOL state (BCM5708 and BCM5709S only) <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 3: UMP Interface mode (BCM5708, BCM5709 and BCM5716 only) <ul style="list-style-type: none"> - 0: MII - 1: RMII • Bit 2: Allow power consumption to exceed 375 mA in VAUX Current Overdraw—Enabling this setting allows the power consumption to exceed 375 mA in VAUX. This setting must be enabled to allow proper operation of management firmware such as IPMI and UMP. <ul style="list-style-type: none"> - 0: Disable - 1: Enable <p>Note: This bit should only be enabled on LOM designs.</p> • Bit 1: Port Swap (BCM5709 and BCM5716 only) <ul style="list-style-type: none"> - 0: Disabled - 1: Enabled • Bit 0: Design type <ul style="list-style-type: none"> - 0: NIC design - 1: LOM design
0x128	4	<p>Shared Hardware Configuration 2</p> <ul style="list-style-type: none"> • Bits 31–24: Reserved • Bits 23–0: The size of the NVRAM in bytes (bits 11 to 0 must be 0).

Table 7: Manufacturing Information Region (Cont.)

Offset	Size (Bytes)	Description
0x12c	4	ECO Configuration (BCM5708 B2 and later only). The value of this location is written to offset 0x7c of the EPB to enable specific hardware fixes. Consult with Broadcom before setting any bits in this NVRAM location.
0x130	4	Reserved
0x134	8	Primary Port MAC address. The upper 2 bytes are not used and must be 0.
0x13c	4	Primary Port Hardware Configuration <ul style="list-style-type: none"> • Bits 31-28: Reserved • Bits 27-24: LED Mode (BCM5709 only) <ul style="list-style-type: none"> - 0x0: MAC Mode 1 - 0x1: PHY Mode 1 - 0x2: PHY Mode 2 - 0x3: PHY Mode 3 - 0x4: MAC Mode 2 - 0x5: PHY Mode 4 - 0x6: PHY Mode 5 - 0x7: PHY Mode 6 - 0x8: MAC Mode 3 - 0x9: PHY Mode 7 - 0xa: PHY Mode 8 - 0xb: PHY Mode 9 - 0xc: MAC Mode 4 - 0xd: PHY Mode 10 - 0xe: PHY Mode 11 - 0xf: Reserved • Bits 20–16: Default Link Setting (BCM5708S and BCM5709S only) <ul style="list-style-type: none"> - 0x0: Auto negotiation - 0x1-0x2: Reserved - 0x3: 1G - 0x4: 2.5G - 0x5–0x12: Reserved - 0x13: Auto negotiation with fallback to 1 Gbps - 0x14: Auto negotiation with fallback to 2.5 Gbps - 0x15: Reserved • Bits 15–0: Pre-emphasis and current driver values (BCM5808S and BCM5709S only).
0x140	20	Reserved
0x154	2	Primary Port PCI Vendor ID (BCM5709 and BCM5716 only)
0x156	8	Primary Port PCI Device ID (BCM5709)
0x158	2	Primary Port PCI Subsystem Device ID (BCM5709 and BCM5716 only)
0x15a	2	Primary Port PCI Subsystem Vendor ID (BCM5709 and BCM5716 only)
0x15c	8	Primary Port iSCSI MAC address. The upper 2 bytes are not used and must be 0.
0x164	8	Primary Port Backup permanent L2 MAC address. The upper 2 bytes are not used and must be 0.
0x16c	28	Reserved

Table 7: Manufacturing Information Region (Cont.)

Offset	Size (Bytes)	Description
0x188	8	Secondary Port MAC Address (BCM5709 and BCM5716 only). The upper 2 bytes are not used and must be 0.
0x190	4	Secondary Port Hardware Configuration (BCM5709 and BCM5716 only). (See the definition of the Primary Port Hardware Configuration at offset 0x13c.)
0x194	20	Reserved
0x1a8	2	Secondary Port PCI Vendor ID (BCM5709 and BCM5716 only)
0x1aa	2	Secondary Port PCI Device ID (BCM5709 and BCM5716 only)
0x1ac	2	Secondary Port PCI Subsystem Device ID (BCM5709 and BCM5716 only)
0x1ae	2	Secondary Port PCI Subsystem Vendor ID (BCM5709 and BCM5716 only)
0x1b0	8	Secondary Port iSCSI MAC Address (BCM5709 and BCM5716 only). The upper 2 bytes are not used and must be 0.
0x1b8	8	Secondary Port Backup permanent L2 MAC address (BCM5709 and BCM5716 only). The upper 2 bytes are not used and must be 0.
0x1c0	28	Reserved
0x1dc	32	PCI Express power budget values.
0x1fc	4	CRC-32 of the Manufacturing Information (offset 0x100 to 0x1fb inclusive).

FEATURE CONFIGURATION INFORMATION

This region contains feature configuration information and its total size is 256 bytes. Its format is shown in [Table 8](#). All reserved fields should be written as 0.

Table 8: Feature Configuration Information Region

Offset	Size (Bytes)	Description
0x200	4	Feature configuration information table revision: <ul style="list-style-type: none"> Bits 31–28: Feature configuration Information table revision in ASCII. The current version is 'A.' Bits 27–0: Reserved
0x204	16	Configuration: <ul style="list-style-type: none"> Bits 31–1: Reserved Bit 0: Enable Linux iSCSI in addition to license validation. <ul style="list-style-type: none"> 0: Disabled 1: Enabled
0x208	12	Reserved.
0x214	4	Port Feature configuration: <ul style="list-style-type: none"> Bits 31–28: Reserved Bit 27: Remote PHY Support (BCM5708S and BCM5709S only) <ul style="list-style-type: none"> 0: Disable 1: Enable Bit 26: Management firmware load <ul style="list-style-type: none"> 0: Management firmware load disabled 1: Management firmware load enabled Bit 25: MBA enable: <ul style="list-style-type: none"> 0: MBA host software load disabled 1: MBA host software load enabled

Table 8: Feature Configuration Information Region (Cont.)

Offset	Size (Bytes)	Description
		<ul style="list-style-type: none"> • Bit 24: WOL Enable: <ul style="list-style-type: none"> - 0: WOL Disabled - 1: WOL Enabled • Bits 23–4: Reserved. • Bits 3–0: PCI bar 1 size <ul style="list-style-type: none"> - 0x0: Disabled - 0x1: 64 KB - 0x2: 128 KB - 0x3: 256 KB - 0x4: 512 KB - 0x5: 1 MB - 0x6: 2 MB - 0x7: 4 MB - 0x8: 8 MB - 0x9: 16 MB - 0xa: 32 MB - 0xb: 64 MB - 0xc: 128 MB - 0xd: 256 MB - 0xe: 512 MB - 0xf: 1 GB
0x218	4	<p>WOL Configuration:</p> <ul style="list-style-type: none"> • Bits 31–12: Reserved. • Bit 11: Asymmetric pause capability for auto-negotiation: <ul style="list-style-type: none"> - 0: Advertise not asymmetric pause capable - 1: Advertise asymmetric pause capable • Bit 10: Pause capability for auto-negotiation: <ul style="list-style-type: none"> - 0: Advertise not pause capable - 1: Advertise pause capable • Bits 9–8: Reserved. • Bit 7–6: Maximum speed advertised during auto-negotiation. <ul style="list-style-type: none"> - 0x0: Advertise up to 100 Mbps - 0x1: Advertise up to 1000 Mbps <p>Note: The use of “Advertise up to 1000 Mbps” requires that bit 2 of offset 0x124 (VAUX Current Overdraw) be set to enable.</p> <ul style="list-style-type: none"> - 0x2: Advertise up to 10 Mbps - 0x3: Reserved • Bits 5–4: Reserved.

Table 8: Feature Configuration Information Region (Cont.)

Offset	Size (Bytes)	Description
		<ul style="list-style-type: none"> • Bits 3–0: WOL link speed: <ul style="list-style-type: none"> - 0x0: Auto-negotiation (10/100 only) - 0x1: 10 Mbps half-duplex - 0x2: 10 Mbps full-duplex - 0x3: 100 Mbps half-duplex - 0x4: 100 Mbps full-duplex - 0x5: Reserved - 0x6: 1000 Mbps full-duplex - 0x7–0xf: Reserved
0x21c	4	MBA configuration: <ul style="list-style-type: none"> • Bits 31–22: Reserved. • Bits 21–20: BIOS bootstrap method: <ul style="list-style-type: none"> - 00b: Auto-detect - 01b: BBS (BIOS boot specification) - 10b: INT 18h - 11b: INT 19h • Bits 19–16: Message timeout (in seconds). • Bits 15–8: Expansion ROM size: <ul style="list-style-type: none"> - 0x0: Disabled - 0x1: 1 KB - 0x2: 2 KB - 0x3: 4 KB - 0x4: 8 KB - 0x5: 16 KB - 0x6: 32 KB - 0x7: 64 KB - 0x8: 128 KB - 0x9: 256 KB

Table 8: Feature Configuration Information Region (Cont.)

Offset	Size (Bytes)	Description
0x21c	4	<ul style="list-style-type: none"> - 0xa: 512 KB - 0xb: 1 MB - 0xc: 2 MB - 0xd: 4 MB - 0xe: 8 MB - 0xf: 16 MB • Bit 7: Hot key selection: <ul style="list-style-type: none"> - 0: CTRL-S - 1: CTRL-B • Bit 6: MBA setup prompt: <ul style="list-style-type: none"> - 0: MBA setup prompt disabled - 1: MBA setup prompt enabled • Bits 5–2: MBA Link Speed: <ul style="list-style-type: none"> - 0x0: Auto-negotiate - 0x1: 10 Mbps half-duplex - 0x2: 10 Mbps full-duplex - 0x3: 100 Mbps half-duplex - 0x4: 100 Mbps full-duplex - 0x5: Reserved - 0x6: 1000 Mbps full-duplex - 0x7: Reserved - 0x8: 2500 Mbps full-duplex - 0x9–0xf: Reserved • Bits 1–0: MBA Boot Protocol: <ul style="list-style-type: none"> - 00b: PXE - 01b: RPL - 10b: BOOTP - 11b: iSCSI Boot
0x220	4	<ul style="list-style-type: none"> UMP configuration: <ul style="list-style-type: none"> • Bits 31–6: Reserved • Bits 5–4: UMP ID (BCM5709 and BCM5716 only) • Bit 3: UMP ID NVRAM Enable (BCM5709 and BCM5716) <ul style="list-style-type: none"> - 0: Disabled. SPIO7 and 8 are sampled for the UMP ID. - 1: Enabled. The UMP ID in bits 5–4 is read from NVRAM. • Bit 2: Reserved • Bit 1: Enable UMP firmware to operate in loopback mode <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 0: Reserved
0x224	4	<ul style="list-style-type: none"> MBA VLAN configuration: <ul style="list-style-type: none"> • Bits 31–17: Reserved • Bit 16: VLAN Enable <ul style="list-style-type: none"> - 0: Disable VLAN - 1: Enable VLAN • Bits 15–0: VLAN Value

Table 8: Feature Configuration Information Region (Cont.)

Offset	Size (Bytes)	Description
0x228	1	<p>Resources allocation:</p> <ul style="list-style-type: none"> • Bits 31–28: Version field. The current revision is "A". • Bits 27–2: Reserved • Bit 1: TOE resources (BCM5709 only) <ul style="list-style-type: none"> - 0: IPv4 - 1: IPv6 • Bit 0: iSCSI resources (BCM5709 only) <ul style="list-style-type: none"> - 0: IPv4 - 1: IPv6
0x229	3	Reserved.
0x22c	4	<p>Resource configuration set by the user. When a bit is set in this field, it indicates that the user's pre-allocated chip resources for that particular technology.</p> <ul style="list-style-type: none"> • Bit 31: First-Come-First-Serve (FCFS) Resource Allocation <ul style="list-style-type: none"> - 0: FCFS is enabled - 1: FCFS is disabled • Bits 30–5: Reserved • Bit 4: RDMA <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 3: iSCSI <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 2: Layer 2 Traffic/TOE <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 1: Diagnostic <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 0: Valid <ul style="list-style-type: none"> - 0: Invalid - 1: Valid

Table 8: Feature Configuration Information Region (Cont.)

Offset	Size (Bytes)	Description
0x230	4	Resource configuration presented to the bus driver to enumerate clients. <ul style="list-style-type: none"> • Bits 31–5: Reserved • Bit 4: RDMA <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 3: iSCSI <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 2: Layer 2 traffic/TOE <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 1: Diagnostic <ul style="list-style-type: none"> - 0: Disable - 1: Enable • Bit 0: Valid <ul style="list-style-type: none"> - 0: Invalid - 1: Valid
0x234	2	The number of reserved RDMA connections.
0x236	2	The number of reserved TOE connections.
0x238	2	The number of reserved iSCSI connections.
0x23a	2	The number of reserved iSER connections.
0x23c	2	Reserved
0x23e	2	The number of pending tasks supported by iSCSI.
0x240	4	Reserved
0x244	4	SMBus Configuration <ul style="list-style-type: none"> • Bits 31–8: Reserved • Bits 7–1: SMBus address • Bit 0: Reserved
0x248	4	iSCSI Boot Configuration <ul style="list-style-type: none"> • Bits 31–1: Reserved • Bit 0: Skip Target Boot <ul style="list-style-type: none"> - 0: Skip target boot - 1: Perform target boot
0x24c	60	Reserved
0x288	4	Port Feature Configuration for Secondary Port (see offset 0x214)
0x28c	4	WOL Configuration for Secondary Port (see offset 0x218)
0x290	4	MBA Configuration for Secondary Port (see offset 0x21c)
0x294	4	UMP Configuration for Secondary Port (see offset 0x220)
0x298	4	MBA VLAN Configuration for Secondary Port (see offset 0x224)
0x29c	1	Resource allocation for Secondary Port (see offset 0x228).
0x29d	3	Reserved
0x2a0	4	Resource configuration set by the user for the secondary port (see offset 0x22c).
0x2a4	4	Resource configuration presented to the bus driver to enumerate clients for the secondary port (see offset 0x230).

Table 8: Feature Configuration Information Region (Cont.)

Offset	Size (Bytes)	Description
0x2a8	2	The number of reserved RDMA connections for the secondary port.
0x2aa	2	The number of reserved TOE connections for the secondary port.
0x2ac	2	The number of reserved iSCSI connections for the secondary port.
0x2ae	2	The number of reserved iSER connections for the secondary port.
0x2b0	2	Reserved
0x2b2	2	The number of pending tasks supported by iSCSI for the secondary port.
0x2b4	4	Reserved
0x2b8	4	SMBus Configuration for the secondary port (see offset 0x244).
0x2bc	4	iSCSI Boot Configuration for the secondary port (see offset 0x248)
0x2c0	60	Reserved
0x2fc	4	CRC-32 of the feature configuration information region (offset 0x200 to 0x2fb inclusive).

VITAL PRODUCT DATA

Vital Product Data (VPD) is defined in the PCI 2.2 specification. It contains the following three major regions:

- Product Name String
- VPD-R
- VPD-W

The Broadcom NetXtreme II does not support VPD-W. Product string and VPD-R information are stored at offset 0x300 to 0x3ff of the NVRAM. The offset and length of each field depends on parameters configured during manufacturing process.

Table 9: Vital Product Data Information

Offset	Size (Bytes)	Description
0x300	Up to 48 (y)	Read-only field contains NULL-terminated product name string. The following fields are currently supported in the product name string: <ul style="list-style-type: none"> • Product name: The default value is “Broadcom NetXtreme II Ethernet Controller.”
0x300+y	Up to 128	VPD-R—The following fields are currently supported in VPD-R: <ul style="list-style-type: none"> • Part Number (PN): This field can be up to 16-bytes long. The default value is “BCM5706A0.” • Engineering Change (EC): This field can be up to 10 bytes long. The default value is “220197-2.” • Serial Number (SN): This field can be up to 16-bytes long. The default value is “0123456789.” • Manufacturing ID (MN): This field can be up to 4-bytes long. The default value is “14e4.” • Vendor Specific (V0): This field can be up to 16-bytes long and is not created by default.
0x380	128	Reserved.

LICENSING INFORMATION

The licensing information is used to control the number of TOE/RDMA/iSCSI/iSER connections the user is allowed to use on the NetXtreme II and is allocated 512 bytes.

 **Note:** The use of a single license field on a dual port controller such as the BCM5709 indicates that features are licensed on a per-controller basis, not a per-port basis. All features licensed here will apply to all ports.

Table 10: Licensing Information

Offset	Size (Bytes)	Description
0x400	32	Reserved.
0x420	4	Valid key indicator. <ul style="list-style-type: none"> • Bits 31–1: Reserved • Bit 0: Upgrade Key indicator <ul style="list-style-type: none"> - 0: Upgrade key is not valid - 1: Upgrade key is valid
0x424	12	Upgrade key. A 96-bit truncated hash value of the license information.
0x430	1	Upgrade key type. <ul style="list-style-type: none"> - 0 = BCM5706 - 1 = BCM5708 - 2 = BCM5709
0x431	1	Upgrade key version. Currently defined as 0x0.
0x432	1	Upgrade key length (in DWORDs). Currently defined as 10.
0x433	1	Upgrade key OEM ID. <ul style="list-style-type: none"> • 0x00: Broadcom • 0x01–0xff: Reserved
0x434	2	Upgrade key capability mask. <ul style="list-style-type: none"> • Bits 15–5: Reserved • Bit 4: iSER • Bit 3: iSCSI • Bit 2: TOE • Bit 1: User mode RDMA • Bit 0: Reserved
0x436	2	Upgrade key maximum TOE connection allowed.
0x438	2	Reserved (must be 0).
0x43a	2	Upgrade key maximum number of RDMA connections allowed.
0x43c	2	Upgrade key maximum number of iSCSI initiator connections allowed.
0x43e	2	Upgrade key maximum number of iSCSI target connections allowed.
0x440	2	Upgrade key maximum number of iSER initiator connections allowed.
0x442	2	Upgrade key maximum number of iSER target connections allowed.
0x444	12	Reserved (must be 0).
0x450	4	Upgrade key license serial number (unique per OEM).
0x454	2	Reserved (must be 0).
0x456	2	Upgrade key expiration date. <ul style="list-style-type: none"> • 0xffff: No expiration • Otherwise: ([Year-2001] x 12 x 32 + month x 32 + day) (both month and day values are zero based)
0x458	124	Reserved.
0x4d4	8	Reserved.
0x4dc	32	Opaque OEM data (used by BMAPI).
0x4fc	4	Upgrade key CRC (0x420 to 0x4fb inclusive).

Table 10: Licensing Information (Cont.)

Offset	Size (Bytes)	Description
0x500	40	Reserved.
0x528	2	Licensing information signature in ASCII, defined as 'LK'.
0x52a	2	Licensing format revision in ASCII. The current version is 'A'.
0x52c	20	Random shared secret. This value is used to hash the license key. To avoid exposing the secret through a simple NVM dump, each byte is XORed with a fixed value 0x5a.
0x540	52	Manufacturing key (identical to offset 0x420 to 0x453).
0x574	104	Reserved.
0x5dc	32	Opaque OEM data (used by BMAPI).
0xfc	4	Manufacturing key CRC (0x528 to 0x5fb inclusive).

PROGRAM IMAGES

All other software and firmware entities (such as bootcode, UMP firmware, PXE host software, and so forth) are stored in the Program Image Region. All entries in this region are subject to relocation, providing that their corresponding Code Directory entries are appropriately updated. [Table 11](#) shows the program image region.

Table 11: Program Image Region

Offset	Size (Bytes)	Description
0x600	Up to (256,000–512) bytes	This region contains firmware, PXE binary, ASF binary, ASF configuration, and so forth.

CALCULATING THE CRC-32 CHECKSUM

The following algorithm is used to calculate the CRC-32 checksum values through the NVRAM block.

```
typedef unsigned long u32;
#define MANUFACTURING_INFO_SIZE140
#define CRC32_POLYNOMIAL 0xEDB88320
char Manufacturing_Info[ MANUFACTURING_INFO_SIZE ];
void main()
{
    u32 checksum;
    chksum = ~util_gen_crc(Manufacturing_Info,
                           MANUFACTURING_INFO_SIZE - 4, 0xffffffff);
    *((u32 *)&Manufacturing_Info[MANUFACTURING_INFO_SIZE - 4]) = chksum;
}
u32 util_gen_crc (
    char *pcDatabuf, // Pointer to data buffer
    u32 ulDatalen, // Length of data buffer in bytes
    u32 ulCrc_in) // Initial value
{
    char data;
    u32 idx, bit, crc;
    crc = ulCrc_in;
    for (idx = 0; idx < ulDatalen; idx++)
    {
        data = *pcDatabuf++;
        for (bit = 0; bit < 8; bit++, data >>= 1)
        {
            crc = (crc >> 1) ^ (((crc ^ data) & 1) ? CRC32_POLYNOMIAL : 0);
        }
    }
    return crc;
}
```

PROGRAMMING THE NON-VOLATILE MEMORY

The Broadcom NetXtreme II device has a flexible non-volatile memory system that supports Serial Flash through a dedicated four-signal SPI interface.



Note: For a list of the currently supported Serial Flash devices supported by the Broadcom NetXtreme II, refer to the latest Broadcom NetXtreme II data sheet.

Access to the non-volatile memory interface is controlled through internal configuration, command, and status registers in the NVRAM Register block (see [“Non-Volatile Memory \(NVM\) Registers” on page 470](#)). The NVRAM can be accessed with automated 32-bit read and write commands or configured for bit-bang operation through the NVM control registers. A semaphore register allows up to four software entities to share access to the NVRAM device.



Note: Request level 0 is the highest priority arbitration request level and is reserved for Broadcom NetXtreme II firmware/bootcode.

PHYSICAL ADDRESSING VERSUS LOGICAL ADDRESSING

The NetXtreme II family supports several different serial Flash devices for NVRAM storage. Each of these NVRAM devices implements a unique combination of sectors, blocks, and pages, which must be addressed to access a single word of data. To support these various addressing schemes the NetXtreme II controller has implemented a logical addressing system that allows all supported devices to be accessed in a consistent manner.

In many cases, the physical and logical addresses are the same, so no translation is necessary, but there are exceptions.

Example: The Atmel® AT45DB011B is organized into 512 pages with 264 bytes per page. It uses nine address bits to reference an individual page (A17 to A9) and nine address bits to access an individual byte within a page (A8 to A0). This means that byte addresses 0 to 263 access actual data while addresses 264 to 511 do not, creating a “hole” in the address map. (The byte address actually wraps around to 0).

To address these “holes” in the NVRAM addressing the NetXtreme II controller supports a logical addressing scheme that allows the higher level driver/application functions to ignore the physical details of the NVRAM device and access it through a single, contiguous range of addresses. The pseudo code in the following section demonstrates how physical and logical addresses are converted.

All of the NVRAM addresses listed in [“Code Directory” on page 36](#) are given as logical addresses. It is up to the application to convert these logical addresses to physical addresses if the NVRAM device requires such translation. The only exception to this rule is the NVRAM address given in [“Bootstrap” on page 36](#). This NVRAM address is given as a physical address to reduce the size and complexity of the ROM code executed by the management processor (MCP) when the NetXtreme II controller completes its power-on reset and performs its initial program load from the bootstrap code in NVRAM.

Address translation is required for the BCM5706 and BCM5708 controllers only. When address translation is used the entire address space of the NVRAM device is used (i.e., there are no data holes between pages) and the NVRAM can be accessed by both the NetXtreme II controller as well as external NVRAM programming devices.

Other members of the NetXtreme II family (including the BCM5709) do not require address translation because the controller automatically maps out the “holes” in the logical address and presents all NVRAM devices as if they use 256 bytes per page. As a result of this behavior, the last 8 bytes of an Atmel 264-byte page will not be used and will remain blank.

NVRAM PSEUDO CODE

The REG_RD(), REG_WR(), and DELAY() routines used in the following code are used as a replacement for the OS-specific routines to read registers, write registers, and delay for a specified number of microseconds.

The NetXtreme II controllers are designed to support certain NVRAM devices entirely through hardware. Support for new NVRAM devices can be added later through software through a process called reconfiguration. During reconfiguration, the software detects whether the connected NVRAM device is supported entirely by hardware or whether it is a new device that requires software support. The software then saves this configuration information for other applications and drivers to use.

This reconfiguration step occurs in the routine nvram_init() in the code below and is normally performed by firmware at power-on reset, but there may be cases where the firmware is not executed before the OEM developed application or driver runs. In these cases, the OEM application or driver must perform the reconfiguration as demonstrated in the code example.

```
#define NVRAM_SIZE          0x200
#define NVRAM_MAGIC          0x669955aa
#define CRC32_RESIDUAL       0xdebb20e3

/* Buffered flash (Atmel: AT45DB011B) specific information */
#define EEPROM_PAGE_BITS     2
#define EEPROM_PHY_PAGE_SIZE(1 << EEPROM_PAGE_BITS)
#define EEPROM_BYTE_ADDR_MASK(EEPROM_PHY_PAGE_SIZE-1)
#define EEPROM_PAGE_SIZE     4
#define EEPROM_TOTAL_SIZE    65536

#define BUFFERED_FLASH_PAGE_BITS9
#define BUFFERED_FLASH_PHY_PAGE_SIZE(1 << BUFFERED_FLASH_PAGE_BITS)
#define BUFFERED_FLASH_BYTE_ADDR_MASK(BUFFERED_FLASH_PHY_PAGE_SIZE-1)
#define BUFFERED_FLASH_PAGE_SIZE264
#define BUFFERED_FLASH_TOTAL_SIZE0x21000

#define SAIFUN_FLASH_PAGE_BITS8
#define SAIFUN_FLASH_PHY_PAGE_SIZE(1 << SAIFUN_FLASH_PAGE_BITS)
#define SAIFUN_FLASH_BYTE_ADDR_MASK(SAIFUN_FLASH_PHY_PAGE_SIZE-1)
#define SAIFUN_FLASH_PAGE_SIZE256
#define SAIFUN_FLASH_BASE_TOTAL_SIZE65536

#define ST_MICRO_FLASH_PAGE_BITS8
#define ST_MICRO_FLASH_PHY_PAGE_SIZE(1 << ST_MICRO_FLASH_PAGE_BITS)
#define ST_MICRO_FLASH_BYTE_ADDR_MASK(ST_MICRO_FLASH_PHY_PAGE_SIZE-1)
#define ST_MICRO_FLASH_PAGE_SIZE256
#define ST_MICRO_FLASH_BASE_TOTAL_SIZE65536

#define BCM5709_FLASH_PAGE_BITS8
#define BCM5709_FLASH_PHY_PAGE_SIZE(1 << BCM5709_FLASH_PAGE_BITS)
#define BCM5709_FLASH_BYTE_ADDR_MASK(BCM5709_FLASH_PHY_PAGE_SIZE-1)
#define BCM5709_FLASH_PAGE_SIZE256
```

```

#define NVRAM_TIMEOUT_COUNT30000
#define FLASHDESC_MAX           64

#define FLASH_STRAP_MASK   (NVM_CFG1_FLASH_MODE | NVM_CFG1_BUFFER_MODE | \
                           NVM_CFG1_PROTECT_MODE | \
                           NVM_CFG1_FLASH_SIZE)

#define FLASH_BACKUP_STRAP_MASK(0xf << 26)

struct flash_spec {
    u32 strapping;
    u32 config1;
    u32 config2;
    u32 config3;
    u32 writel;
#define NV_BUFFERED0x00000001
#define NV_TRANSLATE0x00000002
#define NV_WREN0x00000004
    u32 flags;
    u32 page_bits;
    u32 page_size;
    u32 addr_mask;
    u32 total_size;
    u8 *name;
};

/*****
 * Supported Flash NVRAM device data.
 */
static struct flash_spec flash_table[] =
{
#define BUFFERED_FLAGS(NV_BUFFERED | NV_TRANSLATE)
#define NONBUFFERED_FLAGS(NV_WREN)

    /* Slow EEPROM */
    {0x00000000, 0x40830380, 0x009f0081, 0xa184a053, 0xaf000400,
     BUFFERED_FLAGS, SEEPROM_PAGE_BITS, SEEPROM_PAGE_SIZE,
     SEEPROM_BYTE_ADDR_MASK, SEEPROM_TOTAL_SIZE,
     "EEPROM - slow"},

    /* Expansion entry 0001 */
    {0x08000002, 0x4b808201, 0x00050081, 0x03840253, 0xaf020406,
     NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
     SAIFUN_FLASH_BYTE_ADDR_MASK, 0,
     "Entry 0001"},

    /* Saifun SA25F010 (non-buffered flash) */
    /* strap, cfg1, & writel need updates */
    {0x04000001, 0x47808201, 0x00050081, 0x03840253, 0xaf020406,
     NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
     SAIFUN_FLASH_BYTE_ADDR_MASK, SAIFUN_FLASH_BASE_TOTAL_SIZE*2,
     "Non-buffered flash (128kB)"},

    /* Saifun SA25F020 (non-buffered flash) */
    /* strap, cfg1, & writel need updates */
    {0x0c000003, 0x4f808201, 0x00050081, 0x03840253, 0xaf020406,
     NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
     SAIFUN_FLASH_BYTE_ADDR_MASK, SAIFUN_FLASH_BASE_TOTAL_SIZE*4,
     "Non-buffered flash (256kB)"},
```



```

/* Expansion entry 0100 */
{0x11000000, 0x53808201, 0x00050081, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
 SAIFUN_FLASH_BYTE_ADDR_MASK, 0,
 "Entry 0100"},

/* Entry 0101: ST M45PE10 (non-buffered flash, TetonII B0) */
{0x19000002, 0x5b808201, 0x000500db, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, ST_MICRO_FLASH_PAGE_BITS, ST_MICRO_FLASH_PAGE_SIZE,
 ST_MICRO_FLASH_BYTE_ADDR_MASK, ST_MICRO_FLASH_BASE_TOTAL_SIZE*2,
 "Entry 0101: ST M45PE10 (128kB non-buffered)"},

/* Entry 0110: ST M45PE20 (non-buffered flash) */
{0x15000001, 0x57808201, 0x000500db, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, ST_MICRO_FLASH_PAGE_BITS, ST_MICRO_FLASH_PAGE_SIZE,
 ST_MICRO_FLASH_BYTE_ADDR_MASK, ST_MICRO_FLASH_BASE_TOTAL_SIZE*4,
 "Entry 0110: ST M45PE20 (256kB non-buffered)"},

/* Saifun SA25F005 (non-buffered flash) */
/* strap, cfg1, & writel need updates */
{0x1d000003, 0x5f808201, 0x00050081, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
 SAIFUN_FLASH_BYTE_ADDR_MASK, SAIFUN_FLASH_BASE_TOTAL_SIZE,
 "Non-buffered flash (64kB)"},

/* Fast EEPROM */
{0x22000000, 0x62808380, 0x009f0081, 0xa184a053, 0xaf000400,
 BUFFERED_FLAGS, SEEPROM_PAGE_BITS, SEEPROM_PAGE_SIZE,
 SEEPROM_BYTE_ADDR_MASK, SEEPROM_TOTAL_SIZE,
 "EEPROM - fast"},

/* Expansion entry 1001 */
{0x2a000002, 0xb808201, 0x00050081, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
 SAIFUN_FLASH_BYTE_ADDR_MASK, 0,
 "Entry 1001"},

/* Expansion entry 1010 */
{0x26000001, 0x67808201, 0x00050081, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
 SAIFUN_FLASH_BYTE_ADDR_MASK, 0,
 "Entry 1010"},

/* ATMEL AT45DB011B (buffered flash) */
{0x2e000003, 0xe808273, 0x00570081, 0x68848353, 0xaf000400,
 BUFFERED_FLAGS, BUFFERED_FLASH_PAGE_BITS, BUFFERED_FLASH_PAGE_SIZE,
 BUFFERED_FLASH_BYTE_ADDR_MASK, BUFFERED_FLASH_TOTAL_SIZE,
 "Buffered flash (128kB)"},

/* Expansion entry 1100 */
{0x33000000, 0x73808201, 0x00050081, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
 SAIFUN_FLASH_BYTE_ADDR_MASK, 0,
 "Entry 1100"},

/* Expansion entry 1101 */
{0x3b000002, 0x7b808201, 0x00050081, 0x03840253, 0xaf020406,
 NONBUFFERED_FLAGS, SAIFUN_FLASH_PAGE_BITS, SAIFUN_FLASH_PAGE_SIZE,
 SAIFUN_FLASH_BYTE_ADDR_MASK, 0,
 "Entry 1101"},

/* Atmel Expansion entry 1110 */
{0x37000001, 0x76808273, 0x00570081, 0x68848353, 0xaf000400,
 BUFFERED_FLAGS, BUFFERED_FLASH_PAGE_BITS, BUFFERED_FLASH_PAGE_SIZE,
 BUFFERED_FLASH_BYTE_ADDR_MASK, 0,
 "Entry 1110 (Atmel)"},
```

```

/* ATMEL AT45DB021B (buffered flash) */
{0x3f000003, 0x7e808273, 0x00570081, 0x68848353, 0xaf000400,
 BUFFERED_FLAGS, BUFFERED_FLASH_PAGE_BITS, BUFFERED_FLASH_PAGE_SIZE,
 BUFFERED_FLASH_BYTE_ADDR_MASK, BUFFERED_FLASH_TOTAL_SIZE*2,
 "Buffered flash (256kB)" },
};

/*
 * The BCM5709/BCM5716 controllers transparently handle the
 * differences between Atmel 264 byte pages and all
 * flash devices which use 256 byte pages, so no
 * logical-to-physical mapping is required in the
 * driver.
 */
static struct flash_spec flash_5709 = {
    .flags      = NV_BUFFERED,
    .page_bits= BCM5709_FLASH_PAGE_BITS,
    .page_size= BCM5709_FLASH_PAGE_SIZE,
    .addr_mask= BCM5709_FLASH_BYTE_ADDR_MASK,
    .total_size= BUFFERED_FLASH_TOTAL_SIZE * 2,
    .name     = "5709 Buffered flash (256kB)",
};

#define NVM_COMMAND          0x000006400
#define NVM_COMMAND_RST      (1L<<0)
#define NVM_COMMAND_DONE     (1L<<3)
#define NVM_COMMAND_DOIT     (1L<<4)
#define NVM_COMMAND_WR       (1L<<5)
#define NVM_COMMAND_ERASE    (1L<<6)
#define NVM_COMMAND_FIRST    (1L<<7)
#define NVM_COMMAND_LAST     (1L<<8)
#define NVM_COMMAND_WREN     (1L<<16)
#define NVM_COMMAND_WRDI     (1L<<17)
#define NVM_COMMAND_EWSR     (1L<<18)
#define NVM_COMMAND_WRSR     (1L<<19)

#define NVM_STATUS           0x000006404
#define NVM_STATUS_PI_FSM_STATE(0xfL<<0)
#define NVM_STATUS_EE_FSM_STATE(0xfL<<4)
#define NVM_STATUS_EQ_FSM_STATE(0xfL<<8)

#define NVM_WRITE             0x000006408
#define NVM_WRITE_NVM_WRITE_VALUE(0xffffffffL<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_BIT_BANG(0L<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_EECLK(1L<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_EEDATA(2L<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_SCLK(4L<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_CS_B(8L<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_SO(16L<<0)
#define NVM_WRITE_NVM_WRITE_VALUE_SI(32L<<0)

#define NVM_ADDR              0x00000640c
#define NVM_ADDR_NVM_ADDR_VALUE(0xffffffffL<<0)
#define NVM_ADDR_NVM_ADDR_VALUE_BIT_BANG(0L<<0)
#define NVM_ADDR_NVM_ADDR_VALUE_EECLK(1L<<0)

```



```

#define NVM_ADDR_NVM_ADDR_VALUE_EEDATA(2L<<0)
#define NVM_ADDR_NVM_ADDR_VALUE_SCLK(4L<<0)
#define NVM_ADDR_NVM_ADDR_VALUE_CS_B(8L<<0)
#define NVM_ADDR_NVM_ADDR_VALUE_SO(16L<<0)
#define NVM_ADDR_NVM_ADDR_VALUE_SI(32L<<0)

#define NVM_READ                                0x000006410
#define NVM_READ_NVM_READ_VALUE(0xffffffffL<<0)
#define NVM_READ_NVM_READ_VALUE_BIT_BANG(0L<<0)
#define NVM_READ_NVM_READ_VALUE_EECLK(1L<<0)
#define NVM_READ_NVM_READ_VALUE_EEDATA(2L<<0)
#define NVM_READ_NVM_READ_VALUE_SCLK(4L<<0)
#define NVM_READ_NVM_READ_VALUE_CS_B(8L<<0)
#define NVM_READ_NVM_READ_VALUE_SO(16L<<0)
#define NVM_READ_NVM_READ_VALUE_SI(32L<<0)

#define NVM_CFG1                                0x000006414
#define NVM_CFG1_FLASH_MODE          (1L<<0)
#define NVM_CFG1_BUFFER_MODE        (1L<<1)
#define NVM_CFG1_PASS_MODE         (1L<<2)
#define NVM_CFG1_BITBANG_MODE    (1L<<3)
#define NVM_CFG1_STATUS_BIT       (0x7L<<4)
#define NVM_CFG1_STATUS_BIT_FLASH_RDY(0L<<4)
#define NVM_CFG1_STATUS_BIT_BUFFER_RDY(7L<<4)
#define NVM_CFG1_SPI_CLK_DIV     (0xfL<<7)
#define NVM_CFG1_SEE_CLK_DIV      (0x7ffL<<11)
#define NVM_CFG1_PROTECT_MODE    (1L<<24)
#define NVM_CFG1_FLASH_SIZE      (1L<<25)
#define NVM_CFG1_COMPAT_BYPASSS(1L<<31)

#define NVM_CFG2                                0x000006418
#define NVM_CFG2_ERASE_CMD      (0xffL<<0)
#define NVM_CFG2_DUMMY           (0xffL<<8)
#define NVM_CFG2_STATUS_CMD     (0xffL<<16)

#define NVM_CFG3                                0x00000641C
#define NVM_CFG3_BUFFER_RD_CMD (0xffL<<0)
#define NVM_CFG3_WRITE_CMD       (0xffL<<8)
#define NVM_CFG3_BUFFER_WRITE_CMD(0xffL<<16)
#define NVM_CFG3_READ_CMD        (0xffL<<24)

#define NVM_SW_ARB                            0x000006420
#define NVM_SW_ARB_ARB_REQ_SET0(1L<<0)
#define NVM_SW_ARB_ARB_REQ_SET1(1L<<1)
#define NVM_SW_ARB_ARB_REQ_SET2(1L<<2)
#define NVM_SW_ARB_ARB_REQ_SET3(1L<<3)
#define NVM_SW_ARB_ARB_REQ_CLR0(1L<<4)
#define NVM_SW_ARB_ARB_REQ_CLR1(1L<<5)
#define NVM_SW_ARB_ARB_REQ_CLR2(1L<<6)
#define NVM_SW_ARB_ARB_REQ_CLR3(1L<<7)
#define NVM_SW_ARB_ARB_ARBO      (1L<<8)
#define NVM_SW_ARB_ARB_ARB1     (1L<<9)
#define NVM_SW_ARB_ARB_ARB2     (1L<<10)
#define NVM_SW_ARB_ARB_ARB3     (1L<<11)
#define NVM_SW_ARB_ARB_REQ0      (1L<<12)
#define NVM_SW_ARB_ARB_REQ1      (1L<<13)

```

```

#define NVM_SW_ARB_REQ2           (1L<<14)
#define NVM_SW_ARB_REQ3           (1L<<15)

#define NVM_ACCESS_ENABLE          0x00006424
#define NVM_ACCESS_ENABLE_EN      (1L<<0)
#define NVM_ACCESS_ENABLE_WR_EN(1L<<1)

#define NVM_WRITE1                 0x00006428
#define NVM_WRITE1_WREN_CMD        (0xffffL<<0)
#define NVM_WRITE1_WRDI_CMD        (0xffffL<<8)
#define NVM_WRITE1_SR_DATA         (0xffffL<<16)

/*****************************************/
/* Acquire NVRAM lock. */
/*
/* Before the NVRAM can be accessed the caller must acquire an NVRAM lock. */
/* Locks 0 and 2 are reserved, lock 1 is used by firmware and lock 2 is */
/* for use by the driver. */
/*
/* Returns:
/*   0 on success, positive value on failure.
/*****************************************/
static int
acquire_nvram_lock()
{
    u32 val;
    int j, rc = 0;

    /* Request access to the flash interface. */
    REG_WR(NVM_SW_ARB, NVM_SW_ARB_ARB_REQ_SET2);
    for (j = 0; j < NVRAM_TIMEOUT_COUNT; j++) {
        val = REG_RD(NVM_SW_ARB);
        if (val & NVM_SW_ARB_ARB_ARB2)
            break;

        DELAY(5);
    }

    if (j >= NVRAM_TIMEOUT_COUNT) {
        PRINTF("Timeout acquiring NVRAM lock!\n");
        rc = EBUSY;
    }
}

return (rc);
}

/*****************************************/
/* Release NVRAM lock. */
/*
/* When the caller is finished accessing NVRAM the lock must be released. */
/* Locks 0 and 2 are reserved, lock 1 is used by firmware and lock 2 is */
/* for use by the driver. */
/*

```



```

/* Returns:
 *   0 on success, positive value on failure.
 */
static int
release_nvram_lock()
{
    u32 val;
    int j, rc = 0;

    /*
     * Relinquish nvram interface.
     */
    REG_WR(NVM_SW_ARB, NVM_SW_ARB_ARB_REQ_CLR2);

    for (j = 0; j < NVRAM_TIMEOUT_COUNT; j++) {
        val = REG_RD(NVM_SW_ARB);
        if (!(val & NVM_SW_ARB_ARB_ARB2))
            break;

        DELAY(5);
    }

    if (j >= NVRAM_TIMEOUT_COUNT) {
        PRINTF("Timeout releasing NVRAM lock!\n");
        rc = EBUSY;
    }
}

return (rc);
}

/*
 * Enable NVRAM write access.
 */
/*
 * Before writing to NVRAM the caller must enable NVRAM writes.
 */
/*
 * Returns:
 *   0 on success, positive value on failure.
 */
static int
enable_nvram_write()
{
    u32 val;
    int rc = 0;

    val = REG_RD(MISC_CFG);
    REG_WR(MISC_CFG, val | MISC_CFG_NVM_WR_EN_PCI);

    if (!flash_info->buffered) {
        int j;

        REG_WR(NVM_COMMAND, NVM_COMMAND_DONE);
        REG_WR(NVM_COMMAND, NVM_COMMAND_WREN | NVM_COMMAND_DOIT);

        for (j = 0; j < NVRAM_TIMEOUT_COUNT; j++) {
            DELAY(5);

```



```
    val = REG_RD(NVM_COMMAND) ;
    if (val & NVM_COMMAND_DONE)
        break;
}

if (j >= NVRAM_TIMEOUT_COUNT) {
    PRINTF("Timeout writing NVRAM!\n");
    rc = EBUSY;
}
}

return (rc);
}

/*****************************************/
/* Disable NVRAM write access.          */
/*                                     */
/* When the caller is finished writing to NVRAM write access must be */
/* disabled.                         */
/*                                     */
/* Returns:                           */
/*   Nothing.                         */
/*****************************************/
static void
disable_nvram_write()
{
    u32 val;

    val = REG_RD(MISC_CFG);
    REG_WR(MISC_CFG, val & ~MISC_CFG_NVM_WR_EN);
}

/*****************************************/
/* Enable NVRAM access.              */
/*                                     */
/* Before accessing NVRAM for read or write operations the caller must */
/* enabled NVRAM access.            */
/*                                     */
/* Returns:                           */
/*   Nothing.                         */
/*****************************************/
static void
enable_nvram_access()
{
    u32 val;

    val = REG_RD(NVM_ACCESS_ENABLE);
    /* Enable both bits, even on read. */
    REG_WR(NVM_ACCESS_ENABLE,
           val | NVM_ACCESS_ENABLE_EN | NVM_ACCESS_ENABLE_WR_EN);
}
```



```
*****  
/* Disable NVRAM access. */  
/* When the caller is finished accessing NVRAM access must be disabled. */  
/* Returns:  
 * Nothing. */  
*****  
static void  
disable_nvram_access()  
{  
    u32 val;  
  
    val = REG_RD(NVM_ACCESS_ENABLE);  
  
    /* Disable both bits, even after read. */  
    REG_WR(NVM_ACCESS_ENABLE,  
           val & ~(NVM_ACCESS_ENABLE_EN |  
                  NVM_ACCESS_ENABLE_WR_EN));  
}  
  
*****  
/* Erase NVRAM page before writing. */  
/* Non-buffered flash parts require that a page be erased before it is  
written. */  
/* Returns:  
 * 0 on success, positive value on failure. */  
*****  
static int  
nvram_erase_page(u32 offset)  
{  
    u32 cmd;  
    int j, rc = 0;  
  
    /* Buffered flash doesn't require an erase. */  
    if (flash_info->buffered)  
        goto nvram_erase_exit;  
  
    /* Build an erase command. */  
    cmd = NVM_COMMAND_ERASE | NVM_COMMAND_WR |  
          NVM_COMMAND_DOIT;  
  
    /*  
     * Clear the DONE bit separately, set the NVRAM address to erase,  
     * and issue the erase command.  
     */  
    REG_WR(NVM_COMMAND, NVM_COMMAND_DONE);  
    REG_WR(NVM_ADDR, offset & NVM_ADDR_NVM_ADDR_VALUE);  
    REG_WR(NVM_COMMAND, cmd);  
  
    /* Wait for completion. */  
    for (j = 0; j < NVRAM_TIMEOUT_COUNT; j++) {  
        u32 val;
```

```
DELAY(5);

val = REG_RD(NVM_COMMAND);
if (val & NVM_COMMAND_DONE)
    break;
}

if (j >= NVRAM_TIMEOUT_COUNT) {
    PRINTF("Timeout erasing NVRAM.\n");
    rc = EBUSY;
}

nvram_erase_page_exit:
    return (rc);
}

/*****************************************/
/* Read a dword (32 bits) from NVRAM. */
/*
/* Read a 32 bit word from NVRAM. The caller is assumed to have already */
/* obtained the NVRAM lock and enabled the controller for NVRAM access. */
/*
/* Returns:
/* 0 on success and the 32 bit value read, positive value on failure. */
/*****************************************/
static int
nvram_read_dword(u32 offset, u8 *ret_val, u32 cmd_flags)
{
    u32 cmd;
    int i, rc = 0;

    /* Build the command word. */
    cmd = NVM_COMMAND_DOIT | cmd_flags;

    /* Calculate the offset for buffered flash if translation is used. */
    if (flash_info->flags & NV_TRANSLATE) {
        offset = ((offset / flash_info->page_size) <<
                  flash_info->page_bits) +
                  (offset % flash_info->page_size);
    }

    /*
     * Clear the DONE bit separately, set the address to read,
     * and issue the read.
     */
    REG_WR(NVM_COMMAND, NVM_COMMAND_DONE);
    REG_WR(NVM_ADDR, offset & NVM_ADDR_NVM_ADDR_VALUE);
    REG_WR(NVM_COMMAND, cmd);

    /* Wait for completion. */
    for (i = 0; i < NVRAM_TIMEOUT_COUNT; i++) {
        u32 val;

        DELAY(5);
```



```
val = REG_RD(NVM_COMMAND) ;
if (val & NVM_COMMAND_DONE) {
    val = REG_RD(NVM_READ) ;

    /* Convert big-endian format to host format. */
    val = be32toh(val);
    memcpy(ret_val, &val, 4);
    break;
}
}

/* Check for errors. */
if (i >= NVRAM_TIMEOUT_COUNT) {
    PRINTF("Timeout error reading NVRAM at offset 0x%08X!\n", offset);
    rc = EBUSY;
}

return(rc);
}

/*****************************************/
/* Write a dword (32 bits) to NVRAM.      */
/*                                         */
/* Write a 32 bit word to NVRAM. The caller is assumed to have already */
/* obtained the NVRAM lock, enabled the controller for NVRAM access, and */
/* enabled NVRAM write access.           */
/*                                         */
/* Returns:                                */
/*     0 on success, positive value on failure. */
/*****************************************/
static int
nvr昌_write_dword(u32 offset, u8 *val, u32 cmd_flags)
{
    u32 cmd, val32;
    int j, rc = 0;

    /* Build the command word. */
    cmd = NVM_COMMAND_DOIT | NVM_COMMAND_WR | cmd_flags;

    /* Calculate the offset for buffered flash if translation is used. */
    if (flash_info->flags & NV_TRANSLATE) {
        offset = ((offset / flash_info->page_size) <<
                   flash_info->page_bits) +
                  (offset % flash_info->page_size);
    }

    /*
     * Clear the DONE bit separately, convert NVRAM data to big-endian,
     * set the NVRAM address to write, and issue the write command
     */
    REG_WR(NVM_COMMAND, NVM_COMMAND_DONE);
    memcpy(&val32, val, 4);

    /* Convert big-endian format to host formation. */
}
```



```
val32 = htobe32(val32);
REG_WR(NVM_WRITE, val32);
REG_WR(NVM_ADDR, offset & NVM_ADDR_NVM_ADDR_VALUE);
REG_WR(NVM_COMMAND, cmd);

/* Wait for completion. */
for (j = 0; j < NVRAM_TIMEOUT_COUNT; j++) {
    DELAY(5);

    if (REG_RD(NVM_COMMAND) & NVM_COMMAND_DONE)
        break;
}
if (j >= NVRAM_TIMEOUT_COUNT) {
    PRINTF("Timeout error writing NVRAM at offset 0x%08X\n", offset);
    rc = EBUSY;
}

return (rc);
}

/*****************************************/
/* Initialize NVRAM access.           */
/*                                     */
/* Identify the NVRAM device in use and prepare the NVRAM interface to */
/* access that device.               */
/* Returns:                         */
/*     0 on success, positive value on failure.                         */
/*****************************************/
static int
init_nvram()
{
    u32 val;
    int j, entry_count, rc = 0;
    struct flash_spec *flash;

    if (CHIP_NUM() == CHIP_NUM_5709) {
        flash_info = &flash_5709;
        goto init_nvram_get_flash_size;
    }

    /* Determine the selected interface. */
    val = REG_RD(NVM_CFG1);

    entry_count = sizeof(flash_table) / sizeof(struct flash_spec);

    /*
     * Flash reconfiguration is required to support additional
     * NVRAM devices not directly supported in hardware.
     * Check if the flash interface was reconfigured
     * by the bootcode.
     */

    if (val & 0x40000000) {
        /* Flash interface reconfigured by bootcode. */
    }
}
```



```
for (j = 0, flash = &flash_table[0]; j < entry_count;
     j++, flash++) {
    if ((val & FLASH_BACKUP_STRAP_MASK) ==
        (flash->config1 & FLASH_BACKUP_STRAP_MASK)) {
        flash_info = flash;
        break;
    }
}
} else {
    /* Flash interface not yet reconfigured. */
    u32 mask;

    if (val & (1 << 23))
        mask = FLASH_BACKUP_STRAP_MASK;
    else
        mask = FLASH_STRAP_MASK;

    /* Look for the matching NVRAM device configuration data. */
    for (j = 0, flash = &flash_table[0]; j < entry_count; j++, flash++) {

        /* Check if the device matches any of the known devices. */
        if ((val & mask) == (flash->strapping & mask)) {
            /* Found a device match. */
            flash_info = flash;

            /* Request access to the flash interface. */
            if ((rc = acquire_nvram_lock()) != 0)
                return rc;

            /* Reconfigure the flash interface. */
            enable_nvram_access();
            REG_WR(NVM_CFG1, flash->config1);
            REG_WR(NVM_CFG2, flash->config2);
            REG_WR(NVM_CFG3, flash->config3);
            REG_WR(NVM_WRITE1, flash->write1);
            disable_nvram_access();
            release_nvram_lock();

            break;
        }
    }
}

/* Check if a matching device was found. */
if (j == entry_count) {
    flash_info = NULL;
    PRINTF("Unknown Flash NVRAM found!\n");
    rc = ENODEV;
}

init_nvram_get_flash_size:
/* Write the flash config data to the shared memory interface. */
val = REG_RD_IND(shmem_base + SHARED_HW_CFG_CONFIG2);
val &= SHARED_HW_CFG2_NVM_SIZE_MASK;
if (val)
```

```
    flash_size = val;
else
    flash_size = flash_info->total_size;

return rc;
}

/*****************************************/
/* Read an arbitrary range of data from NVRAM. */
/*
/* Prepares the NVRAM interface for access and reads the requested data */
/* into the supplied buffer. */
/*
/* Returns:
/*   0 on success and the data read, positive value on failure.
/*****************************************/
static int
nvram_read(u32 offset, u8 *ret_buf, int buf_size)
{
    int rc = 0;
u32 cmd_flags, offset32, len32, extra;

if (buf_size == 0)
    goto nvram_read_exit;

/* Request access to the flash interface. */
if ((rc = acquire_nvram_lock()) != 0)
    goto nvram_read_exit;

/* Enable access to flash interface */
enable_nvram_access();

len32 = buf_size;
offset32 = offset;
extra = 0;

cmd_flags = 0;

if (offset32 & 3) {
    u8 buf[4];
    u32 pre_len;

    offset32 &= ~3;
    pre_len = 4 - (offset & 3);

    if (pre_len >= len32) {
        pre_len = len32;
        cmd_flags = NVM_COMMAND_FIRST | NVM_COMMAND_LAST;
    }
    else {
        cmd_flags = NVM_COMMAND_FIRST;
    }
}

rc = nvram_read_dword(offset32, buf, cmd_flags);
```



```
if (rc)
    return rc;

memcpy(ret_buf, buf + (offset & 3), pre_len);

offset32 += 4;
ret_buf += pre_len;
len32 -= pre_len;
}

if (len32 & 3) {
    extra = 4 - (len32 & 3);
    len32 = (len32 + 4) & ~3;
}

if (len32 == 4) {
    u8 buf[4];

    if (cmd_flags)
        cmd_flags = NVM_COMMAND_LAST;
    else
        cmd_flags = NVM_COMMAND_FIRST |
                    NVM_COMMAND_LAST;

    rc = nvram_read_dword(offset32, buf, cmd_flags);

    memcpy(ret_buf, buf, 4 - extra);
}
else if (len32 > 0) {
    u8 buf[4];

    /* Read the first word. */
    if (cmd_flags)
        cmd_flags = 0;
    else
        cmd_flags = NVM_COMMAND_FIRST;

    rc = nvram_read_dword(offset32, ret_buf, cmd_flags);

    /* Advance to the next dword. */
    offset32 += 4;
    ret_buf += 4;
    len32 -= 4;

    while (len32 > 4 && rc == 0) {
        rc = nvram_read_dword(offset32, ret_buf, 0);

        /* Advance to the next dword. */
        offset32 += 4;
        ret_buf += 4;
        len32 -= 4;
    }

    if (rc)
        goto nvram_read_locked_exit;
```



```
cmd_flags = NVM_COMMAND_LAST;
rc = nvram_read_dword(offset32, buf, cmd_flags);

memcpy(ret_buf, buf, 4 - extra);
}

nvram_read_locked_exit:
/* Disable access to flash interface and release the lock. */
disable_nvram_access();
release_nvram_lock();

nvram_read_exit:
    return rc;
}

/*****************************************/
/* Write an arbitrary range of data from NVRAM.          */
/*                                                       */
/* Prepares the NVRAM interface for write access and writes the requested */
/* data from the supplied buffer. The caller is responsible for           */
/* calculating any appropriate CRCs.                                     */
/*                                                       */
/* Returns:                                                 */
/*   0 on success, positive value on failure.                   */
/*****************************************/
static int
nvram_write(u32 offset, u8 *data_buf, int buf_size)
{
    u32 written, offset32, len32;
    u8 *buf, start[4], end[4];
    int rc = 0;
    int align_start, align_end;

    buf = data_buf;
    offset32 = offset;
    len32 = buf_size;
    align_start = align_end = 0;

    if ((align_start = (offset32 & 3))) {
        offset32 &= ~3;
        len32 += align_start;
        if ((rc = nvram_read(offset32, start, 4)))
            goto nvram_write_exit;
    }

    if (len32 & 3) {
        if ((len32 > 4) || !align_start) {
            align_end = 4 - (len32 & 3);
            len32 += align_end;
            if ((rc = nvram_read(offset32 + len32 - 4,
                end, 4))) {
                goto nvram_write_exit;
            }
        }
    }
}
```



```
if (align_start || align_end) {
    buf = malloc(len32);
    if (buf == 0) {
        rc = ENOMEM;
        goto nvram_write_exit;
    }

    if (align_start) {
        memcpy(buf, start, 4);
    }

    if (align_end) {
        memcpy(buf + len32 - 4, end, 4);
    }
    memcpy(buf + align_start, data_buf, buf_size);
}

written = 0;
while ((written < len32) && (rc == 0)) {
    u32 page_start, page_end, data_start, data_end;
    u32 addr, cmd_flags;
    int i;
    u8 flash_buffer[264];

    /* Find the page_start addr */
    page_start = offset32 + written;
    page_start -= (page_start % flash_info->page_size);

    /* Find the page_end addr */
    page_end = page_start + flash_info->page_size;

    /* Find the data_start addr */
    data_start = (written == 0) ? offset32 : page_start;

    /* Find the data_end addr */
    data_end = (page_end > offset32 + len32) ?
        (offset32 + len32) : page_end;

    /* Request access to the flash interface. */
    if ((rc = acquire_nvram_lock()) != 0)
        goto nvram_write_exit;

    /* Enable access to flash interface */
    enable_nvram_access();

    cmd_flags = NVM_COMMAND_FIRST;
    if (flash_info->buffered == 0) {
        int j;

        /* Read the whole page into the buffer
         * (non-buffer flash only) */
        for (j = 0; j < flash_info->page_size; j += 4) {
            if (j == (flash_info->page_size - 4)) {
                cmd_flags |= NVM_COMMAND_LAST;
            }
        }
    }
}
```



```
    rc = nvram_read_dword(page_start + j, &flash_buffer[j], cmd_flags);

    if (rc)
        goto nvram_write_locked_exit;

    cmd_flags = 0;
}

}

/* Enable writes to flash interface (unlock write-protect) */
if ((rc = enable_nvram_write()) != 0)
    goto nvram_write_locked_exit;

/* Erase the page */
if ((rc = nvram_erase_page(page_start)) != 0)
    goto nvram_write_locked_exit;

/* Re-enable the write again for the actual write */
enable_nvram_write();

/* Loop to write back the buffer data from page_start to
 * data_start */
i = 0;
if (flash_info->buffered == 0) {
    for (addr = page_start; addr < data_start;
        addr += 4, i += 4) {

        rc = nvram_write_dword(addr, &flash_buffer[i], cmd_flags);

        if (rc != 0)
            goto nvram_write_locked_exit;

        cmd_flags = 0;
    }
}

/* Loop to write the new data from data_start to data_end */
for (addr = data_start; addr < data_end; addr += 4, i++) {
    if ((addr == page_end - 4) ||
        ((flash_info->buffered) &&
         (addr == data_end - 4))) {

        cmd_flags |= NVM_COMMAND_LAST;
    }
    rc = nvram_write_dword(addr, buf, cmd_flags);

    if (rc != 0)
        goto nvram_write_locked_exit;

    cmd_flags = 0;
    buf += 4;
}

/* Loop to write back the buffer data from data_end
 * to page_end */
if (flash_info->buffered == 0) {
```



```

for (addr = data_end; addr < page_end;
     addr += 4, i += 4) {

    if (addr == page_end-4) {
        cmd_flags = NVM_COMMAND_LAST;
    }
    rc = nvram_write_dword(addr, &flash_buffer[i], cmd_flags);

    if (rc != 0)
        goto nvram_write_locked_exit;

    cmd_flags = 0;
}
}

/* Disable writes to flash interface (lock write-protect) */
disable_nvram_write();

/* Disable access to flash interface */
disable_nvram_access();
release_nvram_lock();

/* Increment written */
written += data_end - data_start;
}

goto nvram_write_exit;

nvram_write_locked_exit:
    disable_nvram_write();
    disable_nvram_access();
    release_nvram_lock();

nvram_write_exit:
    if (align_start || align_end)
        free(buf);

    return (rc);
}

/*****************/
/* Verifies that NVRAM is accessible and contains valid data. */
/*
/* Reads the configuration data from NVRAM and verifies that the CRC is */
/* correct. */
/*
/* Returns: */
/* 0 on success, positive value on failure. */
/*****************/
static int
nvram_test()
{
    u32 buf[NVRAM_SIZE / 4];
    u8 *data = (u8 *) buf;
    int rc = 0;

```

```
u32 magic, csum;

/*
 * Check that the device NVRAM is valid by reading
 * the magic value at offset 0.
 */
if ((rc = nvram_read(0, data, 4)) != 0) {
    PRINTF("Unable to read NVRAM!\n");
    goto nvram_test_exit;
}

/*
 * Verify that offset 0 of the NVRAM contains
 * a valid magic number.
 */
magic = be32toh(buf[0]);
if (magic != NVRAM_MAGIC) {
    rc = ENODEV;
    PRINTF("Invalid NVRAM magic value! Expected: 0x%08X, Found: 0x%08X\n",
           NVRAM_MAGIC, magic);
    goto nvram_test_exit;
}

/*
 * Verify that the device NVRAM includes valid
 * configuration data.
 */
if ((rc = nvram_read(0x100, data, NVRAM_SIZE)) != 0) {
    PRINTF("Unable to read Manufacturing Information from NVRAM !\n");
    goto nvram_test_exit;
}

csum = ether_crc32_le(data, 0x100);
if (csum != CRC32_RESIDUAL) {
    rc = ENODEV;
    PRINTF("Invalid Manufacturing Information NVRAM CRC! Expected: 0x%08X, Found:
0x%08X\n",
           CRC32_RESIDUAL, csum);
    goto nvram_test_exit;
}

csum = ether_crc32_le(data + 0x100, 0x100);
if (csum != CRC32_RESIDUAL) {
    rc = ENODEV;
    PRINTF("Invalid Feature Configuration Information NVRAM CRC! Expected: 0x%08X, Found:
0x%08X\n",
           CRC32_RESIDUAL, csum);
}

nvram_test_exit:
    return rc;
}
```



Section 4: Common Data Structures

HOST MEMORY DATA STRUCTURES

The host status block is used by the NetXtreme II controller to pass information about the current hardware state to software and must be aligned on an 8-byte boundary.

HOST STATUS BLOCK (STATUS)

06	08	09	16
X	X	-	-

Table 12: Host Status Block (Status)

	31–24	23–16	15–8	7–0
0x0		status_attn_bits		
0x4		status_attn_bits_ack		
0x8	status_tx_quick_consumer_index0		status_tx_quick_consumer_index1	
0xc	status_tx_quick_consumer_index2		status_tx_quick_consumer_index3	
0x10	status_rx_quick_consumer_index0		status_rx_quick_consumer_index1	
0x14	status_rx_quick_consumer_index2		status_rx_quick_consumer_index3	
0x18	status_rx_quick_consumer_index4		status_rx_quick_consumer_index5	
0x1c	status_rx_quick_consumer_index6		status_rx_quick_consumer_index7	
0x20	status_rx_quick_consumer_index8		status_rx_quick_consumer_index9	
0x24	status_rx_quick_consumer_index10		status_rx_quick_consumer_index11	
0x28	status_rx_quick_consumer_index12		status_rx_quick_consumer_index13	
0x2c	status_rx_quick_consumer_index14		status_rx_quick_consumer_index15	
0x30	status_completion_producer_index		status_cmd_consumer_index	
0x34	status_idx		status_unused	

06	08	09	16
-	-	X	X

Table 13: Status Block for MSI-X (status_msix)

	31–24	23–16	15–8	7–0
0x0		status_attn_bits		
0x4		status_attn_bits_ack		

Table 13: Status Block for MSI-X (*status_msix*) (Cont.)

	31–24	23–16	15–8	7–0
0x8	status_tx_quick_consumer_index0		status_tx_quick_consumer_index1	
0xc	status_tx_quick_consumer_index2		status_tx_quick_consumer_index3	
0x10	status_rx_quick_consumer_index0		status_rx_quick_consumer_index1	
0x14	status_rx_quick_consumer_index2		status_rx_quick_consumer_index3	
0x18	reserved		reserved	
0x1c	reserved		reserved	
0x20	reserved		reserved	
0x24	reserved		reserved	
0x28	reserved		reserved	
0x2c	reserved		reserved	
0x30	status_completion_producer_index		status_cmd_consumer_index	
0x34	status_idx		reserved	status_blk_num
0x38	reserved			
0x3c		reserved		

Host Status Block Attention Bits (status_attn_bits)

This field displays the current value of each attn-bit element. If one of these bit values differs from the value in the status_attn_bits_ack word, then the bit's value is locked until the “[Status Bit Set Command Register \(pcicfg_status_bit_set_cmd, Offset 0x88\)](#)” on page 178 or the “[Status Bit Clear Command Register \(pcicfg_status_bit_clear_cmd, Offset 0x8c\)](#)” on page 178 have been set to acknowledge the current state (either 0 or 1). This locking provides positive handshake between the driver and the generation of the status blocks.

Table 14: Host Status Block Attention Bits (status_attn_bits)

Bit	Name	Description	06	08	09	16
31	PARITY_ERROR	When this bit is set, it indicates that one or more of the internal RAM devices has experienced a Parity error. All I/O activity within the chip has been suspended (except for status block generation) and the chip is waiting for instructions on how to recover from the parity error.	X	X	X	X
30	EPB_ERROR	When this bit is set, it indicates that the PCIe bridge is indicating an abort condition.	-	X	X	X
30	RESERVED		X	-	-	-
29–28	RESERVED		X	X	X	X
27	GRC_ABORT	When this bit is set, it indicates that the GRC has experienced an access from the PCI bus that has timed out.	X	X	X	X
26	FLSH_ABORT	When this bit is set, it indicates that one or more attention conditions from the FLSH has occurred.	X	X	X	X
25	DMAE_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the “ DMA Command Register (dma_command, Offset 0xc00) ” on page 278 has occurred.	X	X	X	X
24	TIMER_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
23	MAC_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the “ EMAC Attention Enable Register (emac_attention_ena, Offset 0x1408) ” on page 316 has occurred.	X	X	X	X
22	MGMT_PROCESSOR_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the “ Non-Volatile Memory (NVM) Registers ” on page 470 has occurred.	X	X	X	X
21	CMD_PROCESSOR_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
20	CMD_SCHEDULER_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
19	CONTEXT_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the “ CTX Command Register (ctx_command, Offset 0x1000) ” on page 292 has occurred.	X	X	X	X
18	MAILBOX_QUEUE_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the “ MQ Command Register (mq_command, Offset 0x3c00) ” on page 403 has occurred.	X	X	X	X

Table 14: Host Status Block Attention Bits (status_attn_bits) (Cont.)

Bit	Name	Description	06	08	09	16
17	HOST_COALESCE_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the "HC Command Register (hc_command, Offset 0x6800)" on page 484 has occurred.	X	X	X	X
16	COMPLETION_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
15	RX_DMA_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
14	RX_BD_CACHE_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
13	RX_V2P_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the "RV2P Command Register (rv2p_command, Offset 0x2800)" on page 382 has occurred.	X	X	X	X
12	RX_PROCESSOR_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
11	RX_LOOKUP_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
10	RX_MBUF_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
9	RX_PARSER_CATCHUP_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
8	RX_PARSER_MAC_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the "RPM Command Register (rpm_command, Offset 0x1800)" on page 341 has occurred.	X	X	X	X
7	TX_ASSEMBLER_ABORT	When this bit is set, it indicates that one or more attention condition has occurred.	X	X	X	X
6	TX_PATCHUP_ABORT	When this bit is set, it indicates that one or more attention conditions occurred.	X	X	X	X
5	TX_DMA_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
4	TX_PROCESSOR_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
3	TX_BD_CACHE_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
2	TX_BD_READ_ABORT	When this bit is set, it indicates that one or more attention conditions listed in the "TBDR Command Register (tbdr_command, Offset 0x5000)" on page 451 has occurred.	X	X	X	X
1	TX_SCHEDULER_ABORT	When this bit is set, it indicates that one or more attention conditions has occurred.	X	X	X	X
0	LINK_STATE	This bit indicates the current link state of the EMAC. A 1 indicates that the MAC is seeing link.	X	X	X	X

Host Status Block Attention Acknowledge (status_attn_bits_ack)

This field is the currently acknowledged value for each bit in the “[Host Status Block Attention Bits \(status_attn_bits\)](#)” on page 75 field, and indicates the last value that was acknowledged by the host driver. If a bit value in this register is different than the value in the status_attn_bits field, then that bit is requesting attention. Once the attention is complete, the status should be acknowledged by setting or clearing the appropriate bit in this field by using the “[Status Bit Set Command Register \(pcicfg_status_bit_set_cmd, Offset 0x88\)](#)” on page 178 or the “[Status Bit Clear Command Register \(pcicfg_status_bit_clear_cmd, Offset 0x8c\)](#)” on page 178.

TX Quick Consumer Index 0 (status_tx_quick_consumer_index0)

These values are the Quick completion value for BDs transmitted on the L2 connection(s), which is used by the driver to determine when records have been read from the TX Legacy BD Chain.

TX Quick Consumer Index 1 (status_tx_quick_consumer_index1)

These values are the Quick completion value for BDs transmitted on the L2 connection(s), which is used by the driver to determine when records have been read from the TX Legacy BD Chain.

TX Quick Consumer Index 2 (status_tx_quick_consumer_index2)

These values are the Quick completion value for BDs transmitted on the L2 connection(s), which is used by the driver to determine when records have been read from the TX Legacy BD Chain.

TX Quick Consumer Index 3 (status_tx_quick_consumer_index3)

These values are the Quick completion value for BDs transmitted on the L2 connection(s), which is used by the driver to determine when records have been read from the TX Legacy BD Chain.

RX Quick Consumer Index 0 (status_rx_quick_consumer_index0)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 1 (status_rx_quick_consumer_index1)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 2 (status_rx_quick_consumer_index2)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 3 (status_rx_quick_consumer_index3)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 4 (status_rx_quick_consumer_index4)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 5 (status_rx_quick_consumer_index5)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 6 (status_rx_quick_consumer_index6)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 7 (status_rx_quick_consumer_index7)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 8 (status_rx_quick_consumer_index8)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 9 (status_rx_quick_consumer_index9)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 10 (status_rx_quick_consumer_index10)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that



has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 11 (status_rx_quick_consumer_index11)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 12 (status_rx_quick_consumer_index12)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 13 (status_rx_quick_consumer_index13)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 14 (status_rx_quick_consumer_index14)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

RX Quick Consumer Index 15 (status_rx_quick_consumer_index15)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled. 16 of these indexes are supported and can be used for any mix of Generic BD Chains and Legacy L2 BD Chains as needed.

Completion Producer Index (status_completion_producer_index)

This value is used by the driver to track when new records have been added to the kernel completion queue. It is incremented once for each new record that has been added to the chain and wraps to 0 when it reaches the size of the kernel completion queue.

Command Consumer Index (status_cmd_consumer_index)

This value is used by the driver to track when records have been processed on the kernel work queue by the Command Processor. It is incremented once for each record that has been processed on the chain and wraps to 0 when it reaches the size of the kernel work queue.

Index (status_idx)

This is the status block index field. This value is incremented by the Host Coalescing block each time a new status block is written. This value is read by the driver and written to the INDEX field of the “[Interrupt Acknowledge Command Register \(pcicfg_int_ack_cmd, Offset 0x84\)](#)” on page 177 to allow the Host Coalescing state machine to know exactly which status block update was last processed by the software.

Block Number (status_blk_num)

This is the number of the status block.

PER PROCESSOR STATUS BLOCK (STATUS_PER_CPU_BLOCK)

The per processor status block (status_per_cpu_block) is repeated 8 times, immediately following the host status block (status) in host physical memory.

06	08	09	16
—	—	X	X

Table 15: Per Processor Status Block (status_per_cpu_block)

	31–24	23–16	15–8	7–0
0x0	status_pcpu_tx_quick_consumer_index		status_pcpu_rx_quick_consumer_index	
0x4	status_pcpu_completion_producer_index		status_pcpu_cmd_consumer_index	
0x8		reserved		
0xc	status_pcpu_idx		reserved	status_pcpu_blk_nu
0x10		reserved		
0x14		reserved		
0x18		reserved		
0x1c		reserved		
0x20		reserved		
0x24		reserved		
0x28		reserved		
0x2c		reserved		
0x30		reserved		
0x34		reserved		
0x38		reserved		
0x3c		reserved		

TX Quick Consumer Index (status_pcpu_tx_quick_consumer_index)

These values are the Quick completion value for BDs transmitted on the L2 connection(s), which is used by the driver to determine when records have been read from the TX Legacy BD Chain.

RX Quick Consumer Index (status_pcpu_rx_quick_consumer_index)

These values are used by the driver to track when records have been either read by the NetXtreme II for consumption from Generic BD Chains, or when Legacy L2 BD Chains have been filled with data. It is incremented once for each record that has been read/filled.

Completion Producer Index (status_pcpu_completion_producer_index) This value is used by the driver to track when new records have been added to the kernel completion queue. It is incremented once for each new record that has been added to the chain and wraps to 0 when it reaches the size of the kernel completion queue.

Command Consumer Index (status_pcpu_cmd_consumer_index) This value is used by the driver to track when records have been processed on the kernel work queue by the Command Processor. It is incremented once for each record that has been processed on the chain and wraps to 0 when it reaches the size of the kernel work queue.

Index (status_pcpu_idx)

This is the status block index field. This value is incremented by the Host Coalescing block each time a new status block is written. This value is read by the driver and written to the INDEX field of the “Interrupt Acknowledge Command Register (pcifg_int_ack_cmd, Offset 0x84)” on page 315 to allow the Host Coalescing state machine to know exactly which status block update was last processed by the software.

Block Number (status_pcpu_blk_num)

This is the number of the per processor status block.

HOST STATISTICS BLOCK (STAT)

Table 16: Host Statistics Block (Stat)

	31–24	23–16	15–8	7–0
0x0			stat_IfHCInOctets	
0x4				
0x8			stat_IfHCInBadOctets	
0xc				
0x10			stat_IfHCOutOctets	
0x14				
0x18			stat_IfHCOutBadOctets	
0x1c				
0x20			stat_IfHCInUcastPkts	
0x24				
0x28			stat_IfHCInMulticastPkts	
0x2c				
0x30			stat_IfHCInBroadcastPkts	
0x34				
0x38			stat_IfHCOutUcastPkts	
0x3c				

Table 16: Host Statistics Block (Stat) (Cont.)

	31–24	23–16	15–8	7–0
0x40			stat_IfHCOutMulticastPkts	
0x44				
0x48			stat_IfHCOutBroadcastPkts	
0x4c				
0x50		stat_Dot3StatsInternalMacTransmitErrors		
0x54		stat_Dot3StatsCarrierSenseErrors		
0x58		stat_Dot3StatsFCSErrors		
0x5c		stat_Dot3StatsAlignmentErrors		
0x60		stat_Dot3StatsSingleCollisionFrames		
0x64		stat_Dot3StatsMultipleCollisionFrames		
0x68		stat_Dot3StatsDeferredTransmissions		
0x6c		stat_Dot3StatsExcessiveCollisions		
0x70		stat_Dot3StatsLateCollisions		
0x74		stat_EtherStatsCollisions		
0x78		stat_EtherStatsFragments		
0x7c		stat_EtherStatsJabbers		
0x80		stat_EtherStatsUndersizePkts		
0x84		stat_EtherStatsOversizePkts		
0x88		stat_EtherStatsPktsRx64Octets		
0x8c		stat_EtherStatsPktsRx65Octetsto127Octets		
0x90		stat_EtherStatsPktsRx128Octetsto255Octets		
0x94		stat_EtherStatsPktsRx256Octetsto511Octets		
0x98		stat_EtherStatsPktsRx512Octetsto1023Octets		
0x9c		stat_EtherStatsPktsRx1024Octetsto1522Octets		
0xa0		stat_EtherStatsPktsRx1523Octetsto9022Octets		
0xa4		stat_EtherStatsPktsTx64Octets		
0xa8		stat_EtherStatsPktsTx65Octetsto127Octets		
0xac		stat_EtherStatsPktsTx128Octetsto255Octets		
0xb0		stat_EtherStatsPktsTx256Octetsto511Octets		
0xb4		stat_EtherStatsPktsTx512Octetsto1023Octets		
0xb8		stat_EtherStatsPktsTx1024Octetsto1522Octets		
0xbc		stat_EtherStatsPktsTx1523Octetsto9022Octets		
0xc0		stat_XonPauseFramesReceived		
0xc4		stat_XoffPauseFramesReceived		
0xc8		stat_OutXonSent		
0xcc		stat_OutXoffSent		
0xd0		stat_FlowControlDone		
0xd4		stat_MacControlFramesReceived		
0xd8		stat_XoffStateEntered		

Table 16: Host Statistics Block (Stat) (Cont.)

	31–24	23–16	15–8	7–0
0xdc		stat_IfInFramesL2FilterDiscards		
0xe0		stat_IfInRuleCheckerDiscards		
0xe4		stat_IfInFTQDiscards		
0xe8		stat_IfInMBUFDiscards		
0xec		stat_IfInRuleCheckerP4Hit		
0xf0		Reserved		
0xf4		Reserved		
0xf8		Reserved		
0xfc		Reserved		
0x100		Reserved		
0x104		Reserved		
0x108		Reserved		
0x10c		Reserved		
0x110		Reserved		
0x114		Reserved		
0x118		Reserved		
0x11c		Reserved		
0x120		Reserved		
0x124		Reserved		
0x128		Reserved		
0x12c		Reserved		
0x130		Reserved		
0x134		Reserved		
0x138		Reserved		
0x13c		Reserved		

stat_IfHCInOctets

This statistic is collected from the “[EMAC_ifHCInOctets Register \(emac_rx_stat_ifhcinoctets, Offset 0x1500\)](#)” on page 329 and shows the number of octets received on the interface, including framing characters.

stat_IfHCInBadOctets

This statistic is collected from the “[EMAC_ifHCInBadOctets Register \(emac_rx_stat_ifhcinbadoctets, Offset 0x1504\)](#)” on page 330 and shows the number of octets received on the interface that were dropped inside the MAC for any reason.

stat_IfHCOutOctets

This statistic is collected from the “[EMAC_ifHCOutOctets Register \(emac_tx_stat_ifhcoutoctets, Offset 0x1600\)](#)” on page 334 and shows the number of octets that have been transmitted on the interface.

stat_IfHCOOutBadOctets

This statistic is collected from the “[EMAC ifHCOOutBadOctets Register \(emac_tx_stat_ifhcouthadoctets, Offset 0x1604\)](#)” on page 334 and shows the number of octets that the MAC has committed to transmit, but were never transmitted. This includes packets that have suffered excessive or late collisions.

stat_IfHCInUcastPkts

This statistic is collected from the “[EMAC ifHCInUnicastPkts Register \(emac_rx_stat_ifhcincastpkts, Offset 0x150c\)](#)” on page 330 and shows the number of frames received on the wire that were not dropped due to errors that have Unicast Ethernet destination addresses.

stat_IfHCInMulticastPkts

This statistic is collected from the “[EMAC ifHCInMulticastPkts Register \(emac_rx_stat_ifhcincastpkts, Offset 0x1510\)](#)” on page 330 and shows the number of frames received on the wire that were not dropped due to errors that have multicast Ethernet destination addresses.

stat_IfHCInBroadcastPkts

This statistic is collected from the “[EMAC ifHCInBroadcastPkts Register \(emac_rx_stat_ifhcincastpkts, Offset 0x1514\)](#)” on page 330 and shows the number of frames received on the wire that were not dropped due to errors that have the broadcast Ethernet destination addresses.

stat_IfHCOOutUcastPkts

This statistic is collected from the “[EMAC ifHCOOutUnicastPkts Register \(emac_tx_stat_ifhcouthadoctets, Offset 0x162c\)](#)” on page 337 and shows the number of packets transmitted that have unicast destination addresses.

stat_IfHCOOutMulticastPkts

This statistic is collected from the “[EMAC ifHCOOutMulticastPkts Register \(emac_tx_stat_ifhcouthadoctets, Offset 0x1630\)](#)” on page 337 and shows the number of packets transmitted that have multicast destination addresses.

stat_IfHCOOutBroadcastPkts

This statistic is collected from the “[EMAC ifHCOOutBroadcastPkts Register \(emac_tx_stat_ifhcouthadoctets, Offset 0x1634\)](#)” on page 337 and shows the number of packets transmitted that have the broadcast destination address.

stat_Dot3StatsInternalMacTransmitErrors

This statistic is collected from the “[EMAC dot3StatsInternalMacTransmitErrors Register \(emac_tx_stat_dot3statsinternalmactransmiterrors, Offset 0x1654\)](#)” on page 339 and shows the number of times that a frame was dropped due to internal problems in the transmit buffering system. The value for IfOutErrors is really 0 since no data integrity errors can be detected on transmit.

stat_Dot3StatsCarrierSenseErrors

This statistic is collected from the “[EMAC dot3StatsCarrierSenseErrors Register \(emac_rx_stat_dot3statscarriersenseerrors, Offset 0x1520\)](#)” on page 331 and shows the number of times a false carrier has been detected on the internal PHY device. The PHY indicates this by asserting RXER while RXDV is low when the RXD pins are at a state of 0x0e.

stat_Dot3StatsFCSErrors

This statistic is collected from the “[EMAC dot3StatsFCSErrors Register \(emac_rx_stat_dot3statsfcerrors, Offset 0x1518\)](#)” on page 331 and shows the number of frames received on the wire that have an even number of nibbles, fails FCS check, and are of legal length.

stat_Dot3StatsAlignmentErrors

This statistic is collected from the “[EMAC dot3StatsAlignmentErrors Register \(emac_rx_stat_dot3statsalignmenterrors, Offset 0x151c\)](#)” on page 331 and shows the number of frames received on the wire that have an odd number of nibbles, fails FCS check, and are of legal length.

stat_Dot3StatsSingleCollisionFrames

This statistic is collected from the “[EMAC dot3StatsSingleCollisionFrames Register \(emac_tx_stat_dot3statssinglecollisionframes, Offset 0x1618\)](#)” on page 335 and shows the number of collisions that were followed by successful packet transmits. In other words, this is the number of packets that were transmitted with only one collision.

stat_Dot3StatsMultipleCollisionFrames

This statistic is collected from the “[EMAC dot3StatsMultipleCollisionFrames Register \(emac_tx_stat_dot3statsmultiplecollisionframes, Offset 0x161c\)](#)” on page 336 and shows the number of packets that were transmitted with more than one collision.

stat_Dot3StatsDeferredTransmissions

This statistic is collected from the “[EMAC dot3StatsDeferredTransmissions Register \(emac_tx_stat_dot3statsdeferredtransmissions, Offset 0x1620\)](#)” on page 336 and shows the number of packets that were delayed in transmission because they had to wait for an RX packet to complete.

stat_Dot3StatsExcessiveCollisions

This statistic is collected from the “[EMAC dot3StatsExcessiveCollisions Register \(emac_tx_stat_dot3statsexcessivecollisions, Offset 0x1624\)](#)” on page 337 and shows the number of packets that have been dropped due to having 16 collisions in a row.

stat_Dot3StatsLateCollisions

This statistic is collected from the “[EMAC dot3StatsLateCollisions Register \(emac_tx_stat_dot3statslatecollisions, Offset 0x1628\)](#)” on page 337 and shows the number of packets that have been dropped due to having a collision received after the 64-byte collision window.

stat_EtherStatsCollisions

This statistic is collected from the “[EMAC etherStatsCollisions Register \(emac_tx_stat_etherstatscollisions, Offset 0x1608\)](#)” on page 335 and shows the number of collisions that have been detected on the interface.

stat_EtherStatsFragments

This statistic is collected from the “[EMAC etherStatsFragments Register \(emac_rx_stat_etherstatsfragments, Offset 0x1508\)](#)” on page 330 and shows the number of frames received with less than 64 bytes and a bad FCS.

stat_EtherStatsJabbers

This statistic is collected from the “[EMAC etherStatsJabbers Register \(emac_rx_stat_etherstatsjabbers, Offset 0x1538\)](#)” on page 332 and shows the number of frames received that exceed the programmed MTU size and have bad FCS.

stat_EtherStatsUndersizePkts

This statistic is collected from the “[EMAC etherStatsUnderSizePkts Register \(emac_rx_stat_etherstatsundersizepkts, Offset 0x153c\)](#)” on page 332 and shows the number of frames received that are less than 64 bytes in length.

stat_EtherStatsOversizePkts

This statistic is collected from the “[EMAC dot3StatsFramesTooLong Register \(emac_rx_stat_dot3statsframestoolong, Offset 0x1534\)](#)” on page 332 and shows the number of frames received that exceed the programmed MTU size.

stat_EtherStatsPktsRx64Octets

This statistic is collected form the “[EMAC etherStatsPkts64Octets Register \(emac_rx_stat_etherstatspkts64octets, Offset 0x1540\)](#)” on page 333 and shows the number of good frames received of 64 bytes in size.

stat_EtherStatsPktsRx65Octetsto127Octets

This statistic is collected from the “[EMAC etherStatsPkts65Octetsto127Octets Register \(emac_rx_stat_etherstatspkts65octetsto127octets, Offset 0x1544\)](#)” on page 333 and shows the number of good frames received of 65 bytes to 127 bytes in size.

stat_EtherStatsPktsRx128Octetsto255Octets

This statistic is collected from the “[EMAC etherStatsPkts128OctetsTo255Octets Register \(emac_rx_stat_etherstatspkts128octetsto255octets, Offset 0x1548\)](#)” on page 333 and shows the number of good frames received of 128 bytes to 255 bytes in size.

stat_EtherStatsPktsRx256Octetsto511Octets

This statistic is collected from the “[EMAC etherStatsPkts256OctetsTo511Octets Register \(emac_rx_stat_etherstatspkts256octetsto511octets, Offset 0x154c\)](#)” on page 333 and shows the number of good frames received of 256 bytes to 511 bytes in size.

stat_EtherStatsPktsRx512Octetsto1023Octets

This statistic is collected from the “[EMAC etherStatsPkts512OctetsTo1023Octets Register \(emac_rx_stat_etherstatspkts512octetsto1023octets, Offset 0x1550\)](#)” on page 333 and shows the number of good frames received of 512 bytes to 1023 bytes in size.

stat_EtherStatsPktsRx1024Octetsto1522Octets

This statistic is collected from the “[EMAC etherStatsPkts1024OctetsTo1522Octets Register \(emac_rx_stat_etherstatspkts1024octetsto1522octets, Offset 0x1554\)](#)” on page 333 and shows the number of good frames received of 1024 bytes to 1522 bytes in size.

stat_EtherStatsPktsRx1523OctetsTo9022Octets

This statistic is collected from the “[EMAC etherStatsPkts1523OctetsTo9022Octets Register \(emac_rx_stat_etherstatspkts1523octetsto9022octets, Offset 0x1558\)](#)” on page 334 and shows the number of good frames received of 1523 bytes to 9022 bytes in size.

stat_EtherStatsPktsTx64Octets

This statistic is collected from the “[EMAC etherStatsPkts64Octets Register \(emac_tx_stat_etherstatspkts64octets, Offset 0x1638\)](#)” on page 337 and shows the number of good frames transmitted of 64 bytes in size.

stat_EtherStatsPktsTx65OctetsTo127Octets

This statistic is collected from the “[EMAC etherStatsPkts65OctetsTo127Octets Register \(emac_tx_stat_etherstatspkts65octetsto127octets, Offset 0x163c\)](#)” on page 338 and shows the number of good frames transmitted of 65 bytes to 127 bytes in size.

stat_EtherStatsPktsTx128OctetsTo255Octets

This statistic is collected from the “[EMAC etherStatsPkts128OctetsTo255Octets Register \(emac_tx_stat_etherstatspkts128octetsto255octets, Offset 0x1640\)](#)” on page 338 and shows the number of good frames transmitted of 128 bytes to 255 bytes in size.

stat_EtherStatsPktsTx256OctetsTo511Octets

This statistic is collected from the “[EMAC etherStatsPkts256OctetsTo511Octets Register \(emac_tx_stat_etherstatspkts256octetsto511octets, Offset 0x1644\)](#)” on page 338 and shows the number of good frames transmitted of 256 bytes to 511 bytes in size.

stat_EtherStatsPktsTx512OctetsTo1023Octets

This statistic is collected from the “[EMAC etherStatsPkts512OctetsTo1023Octets Register \(emac_tx_stat_etherstatspkts512octetsto1023octets, Offset 0x1648\)](#)” on page 338 and shows the number of good frames transmitted of 512 bytes to 1023 bytes in size.

stat_EtherStatsPktsTx1024OctetsTo1522Octets

This statistic is collected from the “[EMAC etherStatsPkts1024OctetsTo1522Octets Register \(emac_tx_stat_etherstatspkts1024octetsto1522octets, Offset 0x164c\)](#)” on page 338 and shows the number of good frames transmitted of 1024 bytes to 1522 bytes in size.

stat_EtherStatsPktsTx1523OctetsTo9022Octets

This statistic is collected from the “[EMAC etherStatsPkts1522OctetsTo9022Octets Register \(emac_tx_stat_etherstatspkts1523octetsto9022octets, Offset 0x1650\)](#)” on page 338 and shows the number of good frames transmitted of 1523 bytes to 9022 bytes in size.

stat_XonPauseFramesReceived

This statistic is collected from the “[EMAC xonPauseFramesReceived Register \(emac_rx_stat_xonpauseframesreceived, Offset 0x1524\)](#)” on page 331 and shows the number of good MAC control frames received of pause type with a back-off value of 0. This value will increment regardless of flow control state.

stat_XoffPauseFramesReceived

This statistic is collected from the “[EMAC xoffPauseFramesReceived Register \(emac_rx_stat_xoffpauseframesreceived, Offset 0x1528\)](#)” on page 331 and shows the number of good MAC control frames received of pause type with a back-off value other than 0. This value will increment regardless of flow control state.

stat_OutXonSent

This statistic is collected from the “[EMAC outXonSent Register \(emac_tx_stat_outxonsent, Offset 0x160c\)](#)” on page 335 and shows the number of MAC Control pause packets of value 0xffff that have been transmitted.

stat_OutXoffSent

This statistic is collected from the “[EMAC outXoffSent Register \(emac_tx_stat_outxoffsent, Offset 0x1610\)](#)” on page 335 and shows the number of MAC Control pause packets of value 0 that have been transmitted.

stat_FlowControlDone

This statistic is collected from the “[EMAC flowControlDone Register \(emac_tx_stat_flowcontroldone, Offset 0x1614\)](#)” on page 335 and shows the number of times that the transmitter initiates flow control by transmitting the first XOFF packet. This is slightly different from the stat_OutXonSent and stat_OutXoffSent values that count the actual number of pause packets transmitted.

stat_MacControlFramesReceived

This statistic is collected from the “[EMAC macControlFramesReceived Register \(emac_rx_stat_maccontrolframesreceived, Offset 0x152c\)](#)” on page 332 and shows the number of good MAC control frames received that are not of pause type.

stat_XoffStateEntered

This statistic is collected from the “[EMAC xoffStateEntered Register \(emac_rx_stat_xoffstateentered, Offset 0x1530\)](#)” on page 332 and shows the number of times the flow control state has been entered due to the reception of an XOFF pause frame when not in the paused state. The pause flow control mode must be enabled for this counter to increment.

stat_IInFramesL2FilterDiscards

This statistic is collected from the “[RPM L2 Filter Discards Statistic Register \(rpm_stat_l2_filter_discards, Offset 0x1840\)](#)” on page 347 0x1840) and shows the number of good frames that have been dropped due to the L2 perfect match, broadcast, multicast, or MAC Control Frame filters.

stat_IInRuleCheckerDiscards

This statistic is collected from the “[RPM Rule Check Discards Statistic Register \(rpm_stat_rule_checker_discards, Offset 0x1844\)](#)” on page 347 0x1844) and shows the number of good frames dropped due to the rule checker.

stat_IInFTQDiscards

This statistic is collected from the “[RPM iInFtqDiscards Statistic Register \(rpm_stat_ifinftqdiscards, Offset 0x1848\)](#)” on page 347) and shows the number of packets dropped because the FTQ to the RX Lookup block was full. This statistic will only count if there is sufficient storage in the RX MBUF area for the packet. The IInDiscards value is the sum of this value and IInMBUFDiscards



stat_IInMBUFDiscards

This statistic is collected from the “[RPM iInMbufDiscard Statistic Register \(rpm_stat_iinmbufdiscard, Offset 0x184c\)](#)” on page 347 0x184c) and shows the number of packets dropped because the Parser was unable to get sufficient MBUF blocks to store the packet.

stat_IInRuleCheckerP4Hit

This statistic is collected from the “[RPM Rule Checker P4 Hits Statistic Register \(rpm_stat_rule_checker_p4_hit, Offset 0x1850\)](#)” on page 347 0x1850) and shows the number of good frames that have activated each rule in the rule checker.

RECEIVE BUFFER DESCRIPTOR CHAIN RECORD (RX_BD)

Table 17: Receive Buffer Descriptor Chain Record (rx_bd)

	31–24	23–16	15–8	7–0
0x0			rx_bd_haddr	
0x4				
0x8			rx_bd_len	
0xc	Reserved			rx_bd_flags

Host Address Field (rx_bd_haddr)

This value is the host physical address of the buffer where receive data is going to be placed.

Length Field (rx_bd_len)

This value is the length of the host physical buffer in bytes.

Flags Field (rx_bd_flags)

Table 18: Flags Field (rx_bd_flags)

Bit	Name	Description
15–4	RESERVED	
3	START	This bit is set on RX BDs that are at the start of an application based contiguous virtual buffer.
2	END	This bit is set on RX BDs that are at the end of an application based contiguous virtual buffer. The START and END flags may also be used in the Legacy RX BD chain to allow sets of BDs to be maintained as a group. This may be done to handle jumbo packets on systems that cannot allocate a full frame sized physical buffer.
1	DUMMY	This bit should be set in all BD's that describe a dummy application buffer.
0	NOPUSH	This bit is set on all RX BDs that are part of an application buffer that must be filled before it is completed. Buffers for which this bit is set will not be completed if there are remaining empty bytes and a push packet arrives or a push timeout occurs.

TRANSMIT BUFFER DESCRIPTOR CHAIN RECORD (TX_BD)

Table 19: Transmit Buffer Descriptor Chain Record (tx_bd)

	31–24	23–16	15–8	7–0
0x0			tx_bd_haddr	
0x4				
0x8		tx_bd_reserved		tx_bd_nbytes
0xc		tx_bd_vlan_tag		tx_bd_flags

Host Address Field (tx_bd_haddr)

This is the starting physical address in host space for data to be transmitted. If the TX BD Chain Record is the last record on a page, it is referred to as a Next Pointer Record, and this field points to the physical address of the next page in the page chain. The last Next Pointer Record in a page chain will point back to the first page in the page chain.

Reserved Field (tx_bd_reserved)

When TCP segmentation (LSO) is enabled, this field is programmed with the TCP Maximum Segment Size (MSS).

Number of Bytes Field (tx_bd_nbytes)

This value is the number bytes in the physical host data block. This field is reserved if the TX BD Chain Record is a Next Pointer Record.

VLAN Tag Field (tx_bd_vlan_tag)

This VLAN Tag value will be used to create the 32-bit VLAN tag that is inserted at the end of the L2 header. This value is only used on L2 transmits and must be valid on BDs with the START bit and VLAN_TAG bits set. This field is reserved if the TX BD Chain Record is a Next Pointer Record.

Flags Field (tx_bd_flags)

The Flags Field is reserved if the TX BD Chain Record is a Next Pointer Record.

Table 20: Flags Field (tx_bd_flags)

Bit	Name	Description
15	SW_LSO	This bit indicates that this is a TCP segmentation (LSO) request. The MSS is stored in the tx_bd_reserved field. The total TCP + IP headers size in 32-bit words is stored in the lower 4 bits of the SW_OPTION_WORD field.
14	SW_SNAP	This bit indicates that the Ethernet packet is encapsulated in SNAP format.
13	SW_FLAGS	This bit can be used to communicate to custom driver firmware in the TX Processor that this BD needs special processing.
12–8	SW_OPTION_WORD	When the SW_LSO field is set, the lower 4 bits indicate the size of TCP + IP headers in 32-bit words.
7	START	This bit indicates that this BD is the start of an application buffer or the start of an L2 packet.
6	END	The bit indicates that this BD is the end of an application buffer or that this BD completes a L2 packet.

Table 20: Flags Field (tx_bd_flags) (Cont.)

Bit	Name	Description
5	DONT_GEN_CRC	Setting this bit disables the L2 CRC calculation and appending normally done for L2 packets. This is useful for debugging data integrity of the L2 data transfers. This bit is only valid on L2 frames with the START bit set.
4	COAL_NOW	When this bit is set, it will cause the TX Coalescing to force the status block out as soon as this packet is read into the chip. This is used to speed up IPX implementations. This bit is only valid on L2 frames with the START bit set.
3	VLAN_TAG	When this bit is set, it indicates that the value in the tx_bd_vlan_tag field should be used to add a 4 byte VLAN header to the L2 header of the current packet.
2	IP_CKSUM	This bit indicates that the IP header checksum must be calculated and inserted into the packet that starts with this BD. This flag is only valid on L2 frames with the START bit set.
1	TCP_UDP_CKSUM	When this bit is set, it indicates that the TCP or UDP checksum should be calculated and inserted into the appropriate location in the TCP or UDP header. This bit must be set on every tx_bd used by the frame.
0	CONN_FAULT	This bit should never be written for BDs created by any host driver. It is used internally by the Broadcom NetXtreme II.

RX PACKET L2 FRAME STATUS HEADER (L2_FHDR)

Table 21: RX Packet L2 Frame Status Header (l2_fhdr)

	31–24	23–16	15–8	7–0
0x0		l2_fhdr_errors		l2_fhdr_status
0x4			l2_fhdr_hash	
0x8		l2_fhdr_pkt_len		l2_fhdr_vlan_tag
0xc		l2_fhdr_ip_xsum		l2_fhdr_tcp_udp_xsum

Errors Field (l2_fhdr_errors)

Table 22: Errors Field (l2_fhdr_errors)

Bit	Name	Description
15	UDP_BAD_XSUM	This bit is set when the UDP checksum is bad. If the IP_FRAGMENT bit is set then this bit should also be set.
14	RESERVED	
13	TCP_BAD_OFFSET	This bit is set when the offset field of the TCP segment is smaller than the minimum TCP segment size or when it is smaller than the TCP segment.
12	TCP_BAD_XSUM	This bit is set when the TCP checksum is bad. The bit should be set if the TCP_TOO_SHORT bit is set since the length of the TCP payload will not be legal. This bit should also be set for IP fragments since the checksum cannot be valid in this case.
11	TCP_TOO_SHORT	This bit is set when the length of the received frame from the MAC, excluding the L2 and IP header, is less than the minimum TCP segment size of 20 or when the total length indicated by the IP header minus the IP header is less than 20.
10	IP_BAD_XSUM	This bit is set when the IP checksum does not result in a value of 0xffff.



Table 22: Errors Field (I2_fhdr_errors) (Cont.)

Bit	Name	Description
9	IP_BAD_HLEN	This bit is set when the IP header length is less than the minimum size of 20 bytes.
8	IP_BAD_VERSION	This bit is set when the IP version number is not equal to 4.
7	IP_TOO_SHORT	This bit is set when the frame received from the MAC is shorter than 20 bytes or when the total IP length is less than the IP header length.
6	IP_BAD_LEN	This bit is set when the frame received from the MAC, excluding the L2 header, is smaller than the total length field in the IPv4 header.
5	GIANT_FRAME	This bit is set when the frame is longer than the programmed MTU in the MAC.
4	TOO_SHORT	This bit is set when the frame is shorter than 64 bytes.
3	ALIGNMENT	This bit is set when the frame was received with an alignment error. This means that it had an odd number of nibbles and a bad CRC.
2	PHY_DECODE	This bit is set when one more bytes were received with a coding error indication from the PHY.
1	BAD_CRC	This bit is set when an RX frame was received with a bad CRC.
0	RESERVED	—

Status Field (I2_fhdr_status)**Table 23: Status Field (I2_fhdr_status)**

Bit	Name	Description
15	UDP_DATAGRAM	This bit is set when the I2_fhdr_tcp_udp_xsum field is a valid UDP checksum.
14	TCP_SEGMENT	This bit is set when the I2_fhdr_tcp_udp_xsum field is a valid TCP checksum.
13	IP_DATAGRAM	This bit is set when the I2_fhdr_ip_xsum field is a valid IP header checksum.
12–8	L2_HASH	This value indicates the hash of the TCP Tuple value used for spreading a connection load to multiple host processors.
7	L2_LLC_SNAP	This bit is set when the packet was received with LLC_SNAP encapsulation.
6	L2_VLAN_TAG	This bit is set when the packet was received with a VLAN tag. The tag value is in the I2_fhdr_vlan_tag field.
5	RULE_P4	This bit indicates the OR value of all the P4 Rule Checker bits set in rules that were true for this packet. In other words, if any of the P4 rules match this packet, this bit will be set.
4	RULE_P3	This bit indicates the OR value of all the P3 Rule Checker bits set in rules that were true for this packet. In other words, if any of the P4 rules match this packet, this bit will be set.
3	RULE_P2	This bit indicates the OR value of all the P2 Rule Checker bits set in rules that were true for this packet. In other words, if any of the P4 rules match this packet, this bit will be set.
2–0	RULE_CLASS	This bit indicates the lowest Class value of all the Rule Checker rules that were true for this packet.
	Hash Field (I2_fhdr_hash)	This field contains the L2 hash of the TCP tuple used for RSS.
	Packet Length Field (I2_fhdr_pkt_len)	This field contains the packet length.
	VLAN Tag Field (I2_fhdr_vlan_tag)	This field contains the VLAN tag when the L2_VLAN_TAG bit is set.

Table 23: Status Field (I2_fhdr_status) (Cont.)

Bit	Name	Description
	IP Checksum Field (I2_fhdr_ip_xsum)	This field contains the IP checksum when the IP_DATAGRAM bit is set.
	TCP or UDP Checksum Field (I2_fhdr_tcp_udp_xsum)	This field contains the TCP checksum when the TCP_SEGMENT bit is set or the UDP checksum when the UDP_DATAGRAM bit is set.

ON-CHIP MEMORY DATA STRUCTURES

L2 BD CHAIN CONTEXT (L2_BD_CHAIN_CONTEXT)

Table 24: L2 BD Chain Context (l2_bd_chain_context)

	31–24	23–16	15–8	7–0
0x0	ctx_type	ctx_size	bd_pre_read	unused_a
0x4	unused_b			host_bdidx
0x8		host_bseq		
0xc		nx_bseq		
0x10			nx_bdaddr	
0x14				
0x18	unused_c			nx_bdidx

Context Type (ctx_type)

Table 25: Context Type (ctx_type)

Bit	Name	Description
7–4	CTX_BD_CHN_TYPE	This field will be used to identify the L2 BD Chain Context structure type from other Context Structures. This value will be 1 for the context described here.
Bit	Name	Description
0	UNDEFINED	
1	L2_BD_CHN_ID_VAL	All BD Chains Context blocks will have this value set to equal BD_CHN_ID_VAL.
3–0	RESERVED	

BD Miss Pre-Read Count (bd_pre_read)

This value defines the number of BDs to pre-read when a miss is encountered. This value is controlled by firmware and should be optimized during the life of the connection.

Reserved (unused_a)

This byte is currently unused. *The driver or firmware should initialize this byte to a value of 0.*

Reserved (unused_b)

This word is currently unused. *The driver or firmware should initialize this word to a value of 0.*

Host BD Index Mailbox (host_bdidx)

This is the mailbox BD Index write location. The host updates this value as more BDs are added to this BD chain.

Host BD Byte Count Mailbox (host_bseq)

This is the mailbox byte count write location. The host updates this value as more BDs are added to this BD chain.

Next Byte Sequence Pointer (nx_bseq)

This value indicates the corresponding sequence number of the byte that the Next pointer is referencing.

Host Next BD Address Pointer (nx_bdhaddr)

The next four values document the Next pointer used in the data placement algorithm. This value is the Physical host address of BD of the Next location. The Physical Address of the data byte pointed to is boff after the address in the BD that this value points to.

Reserved (unused_c)

This word is currently unused. *The driver or firmware should initialize this word to a value of 0.*

Next BD Index (nx_bdidx)

This is the Next version of the BD Index value that corresponds to the above BD Physical host address.

NEIGHBOR STATE CONTEXT

Table 26: Neighbor State Context

	31–24	23–16	15–8	7–0
0x0	type	size		unused_a
0x4			unused_b	
0x8			unused_c	
0xc			timer1	
0x10		timer2		timer3
0x14		timer4		timer5
0x18	l2hdr_nbytes	flags		da[6]
0x1c			da[6]	
0x20			sa[6]	
0x24		sa[6]		etype
0x28		vlan_tag		ipid_start
0x2c		ipid_count		unused1

Context Type (type)**Table 27: Context Type (type)**

Bit	Name	Description		
		Value	Name	Description
7–4	TYPE	This field will be used to identify the BD Chain Context structure type from other Context Structures.		
		0	EMPTY	—
		1	L2	All L2 Transmit context blocks will have this value.
		2	TCP	All Normal TCP context blocks will have this value.
		3	L5	All L5 context blocks will have this value.
		4	L2_BD_CHN	All RX BD Chain context blocks will have this value.
		5	PG	All PG context blocks will have this value.
3–0	RESERVED			

Context Size (size)

This value defines the length of this context block in 32-byte units.

L2 Header Byte Count (l2hdr_nbytes)

The number of bytes in the L2 header.

Context Flags (flags)**Table 28: Context Flags (flags)**

Bit	Name	Description
7–2	RESERVED	
1	VLAN_TAGGING	PG_VLAN_TAGGING When this bit is set, it indicates that the VLAN tag is expected.
0	SNAP_ENCAP	PG_SNAP_ENCAP When this bit is set, it indicates that the Ethernet frame is using 802.2 SNAP encapsulation.

Ethernet Source Address (sa[6])

The Source MAC address.

Ethernet Type Field (etype)

Ethernet type.

Ethernet VLAN Tag Field (vlan_tag)

VLAN tag.



IP ID Start (ipid_start)

Start IPID for this path.

IP ID Count (ipid_count)

Counter for this path.

L2 CONTEXT (L2_CONTEXT)*Table 29: L2 Context (l2_context)*

	31–24	23–16	15–8	7–0
0x0	type	size		Reserved
0x4–0x87			Reserved	
0x88	cmd_type	est_nbd		tx_host_bidx
0x8c			Reserved	
0x90			tx_host_bseq	
0x94			tsch_bseq	
0x98			tbdr_bseq	
0x9c		tbdr_bidx		tbdr_boff
0xa0			tbdr_bhaddr_hi	
0xa4			tbdr_bhaddr_lo	
0xa8		txp_bidx		txp_boff
0xac			txp_bseq	
0xb0–0xff			Reserved	

Table 30: L2 Context (l2_context)

Bit	Name	Description		
		Value	Name	Description
7–4	TYPE	This field is used to identify the BD chain context structure from other context structures.		
		0	EMPTY	All L2 transmit context blocks have this value.
3–0	RESERVED	1	L2	

size

This value defines the length of this context block in 32-byte units.

tx_host_bidx

est_nbd

cmd_type

tx_host_bseq

tsch_bseq

tbdr_bseq

tbdr_bidx

tbdr_boff

tbdr_bhaddr_hi

tbdr_bhaddr_lo

txp_bidx

txp_boff

txp_bseq



type**Table 31: type**

Bit	Name	Description		
7–4	TYPE	This field will be used to identify the BD Chain Context structure type from other Context Structures.		
		Value	Name	Description
	0	EMPTY	—	
	1	L2	All L2 Transmit context blocks will have this value.	
	2	TCP	All Normal TCP context blocks will have this value.	
	3	L5	All L5 context blocks will have this value.	
3–0	RESERVED	4	L2_BD_CHN	All RX BD Chain context blocks will have this value.

size

This value defines the length of this context block in 32 byte units.

bd_pre_read

This value defines the number of BDs to pre-read when a miss is encountered. This value is controlled by firmware and should be optimized during the life of the connection.

gen_bd_cid

This code defines which BD chain to get new generic buffers from to add to this BD chain. The value is converted to a full 32-bit CID value by shifting it left 7 bits.

gen_bd_max

The maximum allowed number of generic BDs to be used by this connection.

oubits**Table 32: oubits**

Bit	Name	Description	
7	HOST_ACK	—	
6	CP_UPLOAD_COMP	—	
5	COM_TX_UPLOAD	—	
4	COM_RX_UPLOAD	—	
3	TXP_UPLOAD	—	
2	RXP_UPLOAD	—	
1	CP_UPLOAD	—	
0	ACTIVATE	—	

tcp_pgid

This is the pointer to the path-neighbor context block.

tcp_timer1

This is the Retransmit Timer. Write this value with the current value from the timer1 register in the timer block plus an even number of ticks for the desired timeout value. When the timer detects that time as advanced past the value, it will enqueue a command to the Completion Processor. If the Least Significant Bit is 1, then the timer will be disabled.

Table 33: tcp_timer1

Bit	Name	Description
31–1	VALUE	—
0	DISABLE	—

tcp_timer2

This is the RX Push Timer. Write this value with the current value from the timer2 register in the timer block plus an even number of ticks for the desired timeout value. When the timer detects that time as advanced past the value, it will enqueue a command to the RX V2P Block. If the Least Significant Bit is 1, then the timer will be disabled.

Table 34: tcp_timer2

Bit	Name	Description
15–1	VALUE	—
0	DISABLE	—

tcp_timer3

This is the timeout value for timer 3. Write this value with the current value from the timer3 register in the timer block plus an even number of ticks for the desired timeout value. When the timer detects that time as advanced past the value, it will enqueue a command to the Completion Processor. If the Least Significant Bit is 1, then the timer will be disabled.

Table 35: tcp_timer3

Bit	Name	Description
15–1	VALUE	—
0	DISABLE	—

tcp_timer4

This is the timeout value for timer 4. Write this value with the current value from the timer4 register in the timer block plus an even number of ticks for the desired timeout value. When the timer detects that time as advanced past the value, it will enqueue a command to the Completion Processor. If the Least Significant Bit is 1, then the timer will be disabled.

Table 36: tcp_timer4

<i>Bit</i>	<i>Name</i>	<i>Description</i>
15–1	VALUE	—
0	DISABLE	—

tcp_timer5

This is the timeout value for timer 5. Write this value with the current value from the timer5 register in the timer block plus an even number of ticks for the desired timeout value. When the timer detects that time as advanced past the value, it will enqueue a command to the Completion Processor. If the Least Significant Bit is 1, then the timer will be disabled.

Table 37: tcp_timer5

<i>Bit</i>	<i>Name</i>	<i>Description</i>
15–1	VALUE	—
0	DISABLE	—

tcp_snd_wl1

TCP window parameter 1.

tcp_snd_wl2

TCP window parameter 2 * IP, this part of the context is intentionally layout in the order * as the IP header.

tcp_ttl

Time-to-Live (TTL).

tcp_tos

Type of Service (TOS).

tcp_max_adv_win

TCP window parameter 3.

tcp_ip_w3

change: lower 16-bit IP xsum.

tcp_ip_src

Source IP address.

tcp_ip_dst

Destination IP address.

tcp_iphdr_nbytes

* TCP

tcp_snd_seg_scale

TX window scaling.

tcp_rcv_seg_scale

rx window scaling.

tcp_tcp_hlen

Size of TCP header, in # of 32-bit words.

tcp_src_port

Must be together with dst_port AND.

tcp_dst_port

Align on 32-bit boundary.

tcp_mss

This is the Maximum Segment Size.

tcp_flags*Table 38: tcp_flags*

Bit	Name	Description
7	SS	–
6	DISCONNECT	–
5	SEG_SCALING	–
4	SACK	–
3	TIME_STAMP	–
2	NAGLE	–
1	KEEP_ALIVE	When this bit is set, the keep-alive timer will be reset after receiving an incoming segment. When the keep-alive timer is expired, the connection will be uploaded.
0	NO_DELAY_ACK	When this bit is set, there is no delay in sending acknowledgement. Otherwise, the ACK is either sent piggy-back or on expiration of the DELAY_ACK timer.

cp_state

This is the state of the TCP connection.

Table 39: cp_state

Bit	Name	Description		
		Value	Name	Description
7–0	VALUE	Enumeration:		
		0	UNDEFINED	–
		2	LISTEN	–
		4	SYN_SENT	–
		6	SYN_RECV	–
		8	CLOSE_WAIT	–
		10	ESTABLISHED	–
		12	FIN_WAIT1	–
		14	FIN_WAIT2	–
		16	TIME_WAIT	–
		18	CLOSE	–
		20	LAST_ACK	–
		22	CLOSING	–

tcp_rcv_next

This is the current acknowledgement sequence number that goes into the TCP header.

tcp_rcv_next_dack

This is the acknowledgement sequence number that is used to prevent sending of duplicated 0 TCP payload delay ACK.

tcp_rcv_win_seq

This is the last byte in the TCP sequence space expected to receive. The difference between rcv_win_seq and rcv_next is the current advertised receive window.

tcp_snd_una

This is the first byte in the TCP sequence space that its acknowledgement from the peer is still pending.

tcp_snd_next

This is the first byte in the TCP sequence space that will be sent in the next segment from the TX Processor's point of view. This value is shared by the TX DMA and must be passed through the FTQ to the TX DMA so it can optimized write it back to the TX Scheduler if needed.

tcp_snd_max

This is the last byte in the TCP sequence space that was ever sent.

tcp_snd_win

This is the last byte in the TCP sequence space that the peer is expecting. The difference between snd_win and snd_next is the current send window from the TX Processors point of view.

tcp_snd_cwin

This is the last byte in the TCP sequence space that the congestion avoidance algorithm allows to send. Minimum of this and the snd_win values is the effective last byte in the TCP sequence space to be sent.

tcp_tstamp

Timestamp to be echo.

tcp_ssthresh

Slow start threshold.

tcp_sm_rtt

Smoothed Round Trip Time.

tcp_sm_delta

1st order Smoothed Round Trip Time Difference.

tcp_tsch_snd_next

This is the next byte in the TCP sequence space that Tx Scheduler can schedule to send. The difference between this and the minimum of the snd_win_seq and snd_cwin is the current effective send window.

tcp_slot_size*Table 40: tcp_slot_size*

Bit	Name	Description
31	STOP	When this bit is set to 1, the TX Scheduler will remove the connection from its scheduling list and enqueue a record only if prod_retx_num is not equal to tsch_retx_num or if tx_protocol_flags is non-0. This bit allows the TX Processor to stop transmit on a connection for any reason not already accounted for by the TX Scheduler. This bit does not work for L2 connections.
30–24	CMD_MAX	This value is used to wrap the command indexes and is equal to the total number of command cells in the context. This value can never be more than 127. For TCP connections, this value is always 1.
23–0	SLOT_SIZE	This value controls the maximum slot size that will be scheduler for this context. This value may be different from context to context to allow gross shaping of the TX mixing.

tcp_tsch_cmd

This is the consumer command index tracked by the TX Scheduler. This is how it knows what command (from the command cell array) is currently being processed. For TCP connections, this value is always 0.

tcp_cons_retx_num

This is the consumer retransmit number written by the host when all has been updated and is ready to start transmit.

tcp_tsch_xnum*Table 41: tcp_tsch_xnum*

Bit	Name	Description
7	L4	This bit indicates to the TX Scheduler that the connection is L4 and that only one command cell is in use. When this bit is one, the TX Scheduler will do congestion window and TCP window checking. When this bit is 0, the TX Scheduler will do only do congestion window.
6–0	VAL	This value (bottom 7 bits of byte) is incremented once each time a new slot is scheduled by the scheduler so that the TX DMA block can detect when it has caught up to the scheduler and can write back its values to the scheduler, TX Processor, etc.

tcp_tx_protocol_flags

These are the timer flags. By setting a bit in this word, one (or more) timeout conditions down the tx pipeline. The TX Scheduler will clear this byte when it detects that the value is non-0 and it has passed the protocol flags down the tx pipeline in a command.

Table 42: tcp_tx_protocol_flags

Bit	Name	Description
7-6	RESERVED	—
5	FORCE_RST	—
4	LAST_RST	—
3	LAST_ACK	—
2	FORCE_ACK	—
1	UPLOAD	UPLOAD is initiated.
0	TIMER_DELAY_ACK	This bit indicates that the Delay Ack timer has expired.

tcp_prod_retx_num

This value is bumped by completion block when it wants to initiate a retransmit sequence. The completion block should also enqueue a start message to the TX Scheduler after this value has been incremented. This will cause the TX Scheduler to send a retransmit message down the TX Pipeline to flush it. The TX Scheduler should prevent normal transmit when ever this value is not equal to cons_retx_num. It can, however, generate retransmit instructions if tsch_cons_retx_num is not equal to this number.

tcp_tsch_cons_retx_num

This is a local consumer index for the TX Scheduler to allow it to only generate one retransmit message for every increment of prod_retx_num.

tcp_comp_cons_retx_num

This is the completion local consumer index used to allow the completion to send only one retransmit message to the host driver for every increment of prod_retx_num.

tcp_num_retx

Number of times the retransmission has attempted. On receiving a non-0 acknowledgement segment, this field is reset to 0.

tcp_upload_reason

The reason for uploading the connection.

Table 43: tcp_upload_reason

Bit	Name	Description
7	RESERVED	
6	SYN	This bit indicates that a TCP segment with SYN bit set is received on this connection.
5	RST	This bit indicates that a TCP segment with RST bit set is received on this connection.
4	IP_OPTION	This bit indicates that a IP datagram with unknown IP option is received on this connection.
3	FRAGMENT	This bit indicates that a IP fragment is received on this connection.
2	URG	This bit indicates that a TCP segment with urgent data is received on this connection.
1	FIN	REASON_FIN This bit indicates that a TCP segment with FIN bit set is received on this connection.
0	KEEP_ALIVE	REASON_KEEP_ALIVE This bit indicates that the Keepalive timer has expired.

tcp_txp_cmd

This is the command queue index of the command that the TX processor is currently processing.

cmd[1]

One (and only one) command for normal TCP connections is used.

I4_bd_chain_v2p_proc1flags

Flags owned by proc1 that track the state of the BD chain.

Table 44: I4_bd_chain_v2p_proc1flags

Bit	Name	Description
7–1	RESERVED	
0	BD_CHN_FLUSH	This bit indicates that this chain is in the flush state. All packets received on the connection while it is in this state should be dropped.

I4_bd_chain_host_gen_count

A monotonically increasing count of the number of BDs freed by the driver. This count should be incremented whenever a generic BD has been copied into a posted buffer, or when a generic BD has been discarded via an RV2P discard command.

I4_bd_chain_host_bdidx

This is the mailbox BD Index write location. The host will update this value as more BDs are added to this BD chain.

I4_bd_chain_host_bseq

This is the mailbox Bytes Covered write location. The host will update this value as more BDs are added to the BD chain.

I4_bd_chain_nx_bdaddr

The next 4 values document the Next pointer used in the data placement algorithm. This value is the Physical host address of BD of the Next location. The Physical Address of the data byte pointed to is boff after the address in the BD that this value points to.

I4_bd_chain_nx_seq

This value indicates the corresponding sequence number of the byte that the Next pointer is referencing.

I4_bd_chain_v2p_flags

Flags that track the state of the BD chain.

Table 45: I4_bd_chain_v2p_flags

<i>Bit</i>	<i>Name</i>	<i>Description</i>
7–5	RESERVED	
4	PUSH_ARMED	When v2p instructs the completion processor to arm the push timer for a connection, this bit is set. No further arm_push commands will be enqueued until a push timeout occurs, at which point this bit will be cleared.
3	INDICATED	This bit is set when a generic buffer indicate command is sent. No further indicates should be sent until more buffers are posted.
2	HOLE_MODE	This bit is set when the hole_seq pointer is valid. When this bit is set, the area between the io_seq pointer and the hole_seq pointer has not yet been received.
1	GEN_BD_IN_USE	This flag indicates that the BD Chain is currently using generic buffers to track data that has been received on a connection that has not posted buffers yet.
0	RESERVED	

I4_bd_chain_v2p_gen_count

A monotonically increasing count of the number of generic BDs utilized by V2P to store data when posted buffers are not available in this BD chain. The number of generic BDs currently in use is always equal to v2p_gen_count–host_gen_count.

I4_bd_chain_nx_bdidx

This is the Next version of the BD Index value that corresponds to the above BD Physical host address.

I4_bd_chain_gen_used

This number defines the number of bytes that have been consumed in the current generic buffer.

I4_bd_chain_nx_boff

This is the byte offset into area addressed by the above BD and documents the exact byte that is being referenced by the Next pointer.

I4_bd_chain_cmpl_seq

This value indicates the corresponding sequence number of the byte that the Complete pointer is referencing.

I4_bd_chain_cmpl_bdhaddr

The next 4 values document the Complete pointer used in the data placement algorithm. This value is the Physical host address of BD of the Complete location. The Physical Address of the data byte pointed to is boff after the address in the BD that this value points to.

I4_bd_chain_gen_size

This value saves the length of the current generic buffer. If the sequential data received into the current generic buffer exceeds this value, then the current generic buffer will be closed out and a new one will be allocated.

I4_bd_chain_cmpl_bdidx

This is the Complete version of the BD Index value that corresponds to the above BD Physical host address.

I4_bd_chain_io_seq

This value is valid at all times and indicates the sequence number where data has been placed in either generic or posted buffers and all data before this point is placed in order. If there is a hole, this point defines the beginning of the hole.

I4_bd_chain_hole_seq

This value is valid with the HOLE_MODE bit is set. It is the sequence number of the end of the current hole. The beginning of the hole is defined by the io_seq value.

I4_bd_chain_end_seq

This value is the sequence number just past the last byte that has been received. This value is used to determine where to stop completing data at when a hole is filled.

I4_bd_chain_bseq_lead

This value tracks the delta between buffer sequence space and protocol sequence space.

I4_bd_chain_push_seq

This value is the sequence number of the first byte that has been received where the push bit has not been set. All bytes to the left of push_seq are to be pushed into the first available posted buffer.

I4_bd_chain_gen_start_seq

This value documents the boundary between data placed in posted buffers and data posted in generic buffers. If the io_seq value is greater than this number, then the difference is the amount of data available to indicate. This value should never be more than io_seq when generic buffers are in use.

I4_bd_chain_gen_seq

This value is part of the generic buffer support. This value documents the sequence number of first byte in the current generic buffer pointed to by the gen_bfr value.



I4_bd_chain_gen_bfr

This value points to the current generic buffer in use. Only sequential data from the gen_seq value to the gen_used value can be placed in this buffer. of generic buffers have been used.

RX PARSER ACPI-WOL PATTERN FORMAT

Each 64-bit word describes the pattern match information for one byte of packet data. Notice that the 64-bit words are swapped such that byte 0 of the packet data is described by the 2nd 64-bit word of the pattern format, not the 1st. It is also important to note that there is no method for disabling a particular pattern. If no patterns are needed, then the ACPI mode must not be enabled. If less than 7 patterns are needed, then the unused patterns must be filled with duplicate data from one of the used patterns.

Table 46: RX Parser ACPI-WOL Pattern Format

	31–24	23–16	15–8	7–0
0x0	ena1	pattern1_0	pattern1_1	pattern1_2
0x4	pattern1_3	pattern1_4	pattern1_5	pattern1_6
0x8	ena0	pattern0_0	pattern0_1	pattern0_2
0xc	pattern0_3	pattern0_4	pattern0_5	pattern0_6
0x10	ena3	pattern3_0	pattern3_1	pattern3_2
0x14	pattern3_3	pattern3_	pattern3_5	pattern3_6
0x18	ena2	pattern2_0	pattern2_1	pattern2_2
0x1c	pattern2_3	pattern2_	pattern2_5	pattern2_6
0x20–0x3ff		byte[127]		

enaN (N=0,1,2,3)

This byte defines what pattern bytes are enabled for compare.

patternN_0 (N=0,1,2,3)

If ena0[0] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[0] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

patternN_1 (N=0,1,2,3)

If ena0[1] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[1] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

patternN_2 (N=0,1,2,3)

If ena0[2] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[2] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

patternN_3 (N=0,1,2,3)

If ena0[3] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[3] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

patternN_4 (N=0,1,2,3)

If ena0[4] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[4] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

patternN_5 (N=0,1,2,3)

If ena0[5] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[5] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

pattern0_6 to patternN_6 (N=0,1,2,3)

If ena0[6] is set, then this byte will be compared to byte0 of the incoming packet. If the compare fails, the pattern will be disabled from generating an ACPI event. If ena0[6] is clear, then this byte will be checked to see if it is 0xff. If it is, then the pattern will be terminated and if no miss-compare have been found, an ACPI event will be generated.

byte[127]

This is the ACPI Pattern data for the remaining 127 bytes of pattern match. The compare data for each packet byte appears just as it did for byte0 above as a enable byte followed by 7 different pattern bytes. Any pattern without a length indication (enable bit clear with pattern equal to 0xff) will automatically terminate after the 128 byte.

Section 5: Device Control

DEVICE INITIALIZATION CODE CONVENTIONS

Throughout this section, the pseudo code uses several procedures or macros, which implement common functionality such as writing to a register on the NetXtreme II controller. [Table 47](#) summarizes those procedures/macros and their meanings.

Table 47: Procedure Descriptions

Procedure	Description	Comments
REG_WR(bp, register, value)	Writes the value <i>value</i> to the register <i>register</i> in BAR mapped memory space on the device bp.	Common register write access.
REG_RD(bp, register)	Returns the 32-bit value of the register <i>register</i> from BAR mapped memory space of the device bp.	Common register read access.
REG_WR_IND(bp, offset, value)	Writes the 32-bit value <i>value</i> to the offset <i>offset</i> in BAR mapped memory space using indirect memory access on the device bp.	This routine performs register accesses using PCI configuration cycle writes to guarantee strongly ordered data writes.
REG_RD_IND(bp, offset)	Returns the 32-bit value of the offset <i>offset</i> in BAR mapped memory space using indirect memory access on the device bp.	This routine performs register accesses using PCI configuration cycle reads and writes to guarantee strongly ordered data reads.
PCICFG_WR(bp, register, value)	Writes the value <i>value</i> to the PCI configuration register <i>register</i> on the device bp.	This call is often provided as an operating system service.
PCICFG_RD(bp, register)	Returns the value of the PCI configuration register <i>register</i> of the device bp.	This call is often provided as an operating system service.
PHY_WR(bp, register, value)	Writes the value <i>value</i> to the PHY register <i>register</i> on the device bp.	PHY register write access.
PHY_RD(bp, register)	Returns the value of the PHY register <i>register</i> for the device bp.	PHY register read access.
CHIP_NUM(bp)	Returns the part number of the device bp.	Generally used to recognize a group of devices such as all BCM5706 controllers.
CHIP_REV(bp)	Returns the revision of the device bp.	Generally used to identify a stepping of the controller such as the A1 revision part.
CHIP_ID(bp)	Returns both the part number and revision of the device bp.	Generally used to identify a device and stepping in one command such as in the BCM5706 A0.
CTX_WR(bp, cid_addr, offset, value)	Writes the value <i>value</i> to the CID at address <i>cid_addr</i> , starting at offset <i>offset</i> on the device bp.	Uses the “ CTX Data Address Register (ctx_data_adr, Offset 0x1010) ” on page 297 and the “ CTX Data Register (ctx_data, Offset 0x1014) ” on page 298.

Table 47: Procedure Descriptions (Cont.)

Procedure	Description	Comments
CTX_RD(bp, cid_addr, offset)	Returns the value of the CID at address <code>cid_addr</code> , starting at offset <code>offset</code> on the device <code>bp</code> .	Uses the “ CTX Data Address Register (ctx_data_adr, Offset 0x1010) ” on page 297 and the “ CTX Data Register (ctx_data, Offset 0x1014) ” on page 298.
udelay(time)	Forces a delay of <code>time</code> microseconds.	—

Register definitions are given as a combination of the register block and the individual register within that block. For example, the NetXtreme II PCI configuration registers are part of the `pci_config` block, so the “[Vendor ID Register \(pcicfg_vendor_id, Offset 0x00\)](#)” on page 152 is described as `pci_config.pcicfg_vendor_id`. The following register blocks are defined for the NetXtreme II controller:

Table 48: Register Block Naming Convention

Register Block	Block Name
PCI Configuration Block registers	pci_config
Miscellaneous Block registers	misc
Non-Volatile RAM Block registers	nvm
DMA Block registers	dma
Context Block registers	ctx
Ethernet MAC Block registers	emac
Receive MAC Parser Block registers	rpm
Receive Catch-Up Parser Block registers	rpc
Receive Lookup Block registers	rlup
Receive Virtual to Physical Block registers	rv2p
Receive DMA Block registers	rdma
Receive BD Cache Block registers	rbdc
Mailbox Queue Block registers	mq
Command Scheduler Block registers	csch
Timer Block registers	timer
Transmit Scheduler Block registers	tsch
Transmit BD Read Block registers	tbdr
Transmit BD Cache Block registers	tbdc
Transmit DMA Block registers	tdma
Transmit Assembler Block registers	tas
Host Coalescing Block registers	hc

Bit definitions for the individual fields within a register are created by combining the field name along with the register name to create a unique value. For example, the `MASK_INT` field of the “[Interrupt Acknowledge Command Register \(pcicfg_int_ack_cmd, Offset 0x84\)](#)” on page 177 is referred to as `PCICFG_INT_ACK_CMD_MASK_INT` throughout the code to prevent any collisions with duplicate field names.

MAC RESET PROCEDURE

This section lists the steps required to reset the MAC block of the NetXtreme II controller. MAC reset should be performed before initializing the MAC, as part of the host driver unload procedure, and prior to setting the MAC into a low power state. Always refer to the latest errata document for any additional steps required for MAC initialization. This code is based on the bnx2 driver, version 1.1.16, dated March 16th, 2005.

- 1 Wait for any pending operations to complete by setting the TX_DMA_ENABLE, DMA_ENGINE_ENABLE, RX_DMA_ENABLE, and HOST_COALESCING_ENABLE bits of the “[MISC Enable Clear Register \(misc_enable_clr_bits, Offset 0x814\)](#)” on page 234. Delay for five microseconds after writing to the MISC Enable Clear register to allow the state machines to complete any in progress operations.

```
REG_WR(bp, misc.missc_enable_clr_bits,
        MISC_ENABLE_CLR_BITS_TX_DMA_ENABLE |
        MISC_ENABLE_CLR_BITS_DMA_ENGINE_ENABLE |
        MISC_ENABLE_CLR_BITS_RX_DMA_ENABLE |
        MISC_ENABLE_CLR_BITS_HOST_COALESCE_ENABLE) ;
val = REG_RD(bp, misc.missc_enable_clr_bits) ;
udelay(5) ;
```

- 2 Notify the firmware that the driver is about to reset the controller by writing the driver reset signature to the driver firmware mailbox. (See [Section 8: “Driver/Firmware Shared Memory” on page 135](#) for additional details on the driver firmware mailbox.)

```
#define HOST_VIEW_SHMEM_BASE 0x00167c00
#define DRV_RESET_SIGNATURE 0x4841564b /* HAVK */

REG_WR_IND(bp, HOST_VIEW_SHMEM_BASE +
            offsetof(shmem_region_t, drv_fw_mb.drv_reset_signature),
            DRV_RESET_SIGNATURE) ;
```

- 3 Wait for the firmware to acknowledge the driver reset signature. (See [“Firmware Synchronization Pseudo Code” on page 138](#) of [Section 8: “Driver/Firmware Shared Memory”](#) for details information on implementing the driver/firmware mailboxes.)

```
#define DRV_MSG_DATA_WAIT0 0x00010000
fw_sync(bp, DRV_MSG_DATA_WAIT0 | DRV_MSG_CODE_RESET) ;
```

- 4 Perform a dummy read to force the chip to complete all current transactions by reading from the “[MISC ID Register \(misc_id, Offset 0x808\)](#)” on page 229.

```
val = REG_RD(bp, misc.missc_id) ;
```

- 5** Reset the controller by setting the CORE_RST_REQ, REG_WINDOW_ENA, and TARGET_MB_WORD_SWAP bits of the “[Miscellaneous Configuration Register \(pcicfg_misc_config, Offset 0x68\)](#)” on page 172.

```
val = PCICFG_MISC_CONFIG_CORE_RST_REQ |
      PCICFG_MISC_CONFIG_REG_WINDOW_ENA |
      PCICFG_MISC_CONFIG_TARGET_MB_WORD_SWAP;

REG_WR(bp, pci_config.pcicfg_misc_config, val);
```

[See CQ#TO544]

```
if ((CHIP_ID(bp) == CHIP_ID_5706_A0) || (CHIP_ID(bp) == CHIP_ID_5706_A1)) {
    for (i = 0; i < 500; i++) {
        udelay(30);
    }
}
```

- 6** Wait for the controller to complete its reset by polling the CORE_RST_BSY bit of the “[Miscellaneous Configuration Register \(pcicfg_misc_config, Offset 0x68\)](#)” on page 172. The reset should take no longer than 30 µs to complete.

```
for (i = 0; i < 10; i++) {
    val = REG_RD(bp, pci_config.pcicfg_misc_config);
    if ((val & (PCICFG_MISC_CONFIG_CORE_RST_REQ |
                  PCICFG_MISC_CONFIG_CORE_RST_BSY)) == 0) {
        break;
    }
    udelay(10);
}
```

- 7** Wait for the firmware to acknowledge the driver reset signature. (See “[Firmware Synchronization Pseudo Code](#)” on page 138 of [Section 8: “Driver/Firmware Shared Memory”](#) for details in implementing the driver/firmware mailboxes.)

```
fw_sync(bp, DRV_MSG_DATA_WAIT1 | DRV_MSG_CODE_RESET);
```

- 8** Undocumented errata.

```
if (CHIP_ID(bp) == CHIP_ID_5706_A0) {
    /* Adjust the voltage regular to two steps lower. The default
     * of this register is 0x0000000e. */
    REG_WR(bp, misc.misc_vreg_control, 0x000000fa);

    /* Remove bad rbuf memory from the free pool. */
    alloc_bad_rbuf(bp);
}
```

MAC INITIALIZATION PROCEDURE

This section lists the steps required to initialize the MAC block of the NetXtreme II controller. MAC Initialization should always be preceded by MAC reset, which is documented separately. Always refer to the latest errata document for any additional steps required for MAC initialization. This code is based on the bnx2 driver, version 1.1.16, dated March 16th, 2005.

- 1 Disable interrupts by setting the MASK_INT bit of the “[Interrupt Acknowledge Command Register \(pcicfg_int_ack_cmd, Offset 0x84\)](#)” on page 177.

```
REG_WR(bp, pci_config.pcicfg_int_ack_cmd, PCICFG_INT_ACK_CMD_MASK_INT);
```

- 2 Program the “[DMA Configuration Register \(dma_config, Offset 0xc08\)](#)” on page 281.

- a. Enable byte and word swapping for DMA control and data paths. For example, drivers running on little-endian x86 processors should set the DATA_BYTE_SWAP, DATA_WORD_SWAP, and CNTL_WORD_SWAP bits. (See “[Byte-Swapping Issues](#)” on page 128 for more details.)

```
val |= DMA_CONFIG_DATA_BYTE_SWAP | DMA_CONFIG_DATA_WORD_SWAP |
      DMA_CONFIG_CNTL_WORD_SWAP;
```

- b. Set the NO_WCHANS_IN_USE and NO_RCHANS_IN_USE fields to the values of 3 and 5 respectively for better performance.

```
val |= (5 << 12) | (3 << 16);
```

- c. Set the CNTL_PCI_COMP_DLY bit and set the PCI_CLK_CMP_BITS field to 2.

```
val |= (0x2 << 20) | (1 << 11);
```

- d. Errata fix. (Refer to these errata: E9_5706CA1_65, E1_5706CA2_65, E9_5706SA1_65, and E1_5706SA2_65)

```
if ((CHIP_NUM(bp) == CHIP_NUM_5706) &&
    (CHIP_ID(bp) != CHIP_ID_5706_A0) && !(bp->flags & PCIX_FLAG)) {
    val |= DMA_CONFIG_CNTL_PING_PONG_DMA;
}
REG_WR(dma.dma_config, val);
```

- 3 Undocumented errata fix.

```
if (CHIP_ID(bp) == CHIP_ID_5706_A0) {
    val = REG_RD(bp, tdma.tdma_config);
    val |= TDMA_CONFIG_ONE_DMA;
    REG_WR(bp, tdma.tdma_config, val);
}
```



4 Errata fix. (Refer to these errata: E1_5706CA1_562 and E1_5706SA1_562)

```
if (bp->flags & PCIX_FLAG) {
    u16 val16;

    val16 = PCICFG_RD(bp, pci_config.pcicfg_pcix_command);
    val16 &= ~PCICFG_PCIX_COMMAND_RELAX_ORDER;
    PCICFG_WR(bp, pci_config.pcicfg_pcix_command, val16);
}
```

5 Enable the context block (to initialize context memory), the host coalescing block (to provide status block updates), and the RV2P block (to download firmware) by setting the CONTEXT_ENABLE, RX_RV2P_ENABLE, and HOST_COALESCE_ENABLE bits of the “[MISC Enable Set Register \(misc_enable_set_bits, Offset 0x810\)](#)” on page 232.

```
REG_WR(misc.missc_enable_set_bits, MISC_ENABLE_SET_BITS_HOST_COALESCE_ENABLE |
        MISC_ENABLE_STATUS_BITS_RX_V2P_ENABLE |
        MISC_ENABLE_STATUS_BITS_CONTEXT_ENABLE);
```

6 Initialize context memory.

- a. The NetXtreme II implements context memory, which acts as a central storage location for per-connection data. The physical context memory consists of 6,144 entries with 64 bytes per entry (for a total of 384 KB). A context ID (CID) is mapped to physical context memory using a page table, much like an operating system maps virtual memory to physical memory through a page table. The “[CTX Virtual Address Register \(ctx_virt_addr, Offset 0x1008\)](#)” on page 297 links a CID to a page table entry while the “[CTX Page Table Register \(ctx_page_tbl, Offset 0x100c\)](#)” on page 297 links a page table entry to a physical context memory location.
- b. A physical CID is 128 bytes long and maps to 0, 1, or 2 context memory blocks through the page table. A virtual CID is a grouping of physical CIDs and may vary between 128 bytes and 1024 bytes depending on how the context is used.
- c. By default, 96 physical CIDs are pre-allocated by the NetXtreme II firmware for normal L2 driver use, and these are referred to as Quick CIDs. The driver is responsible for clearing these physical CIDs prior to loading firmware for the NetXtreme II controller.

```
#define CTX_SHIFT 7
#define CTX_SIZE (1 << CTX_SHIFT)
#define CTX_MASK (CTX_SIZE - 1)
#define GET_CID_ADDR(_cid) (((_cid) << CTX_SHIFT) | (_cid_addr) >> CTX_SHIFT)

#define PHY_CTX_SHIFT 6
#define PHY_CTX_SIZE (1 << PHY_CTX_SHIFT)
#define PHY_CTX_MASK (PHY_CTX_SIZE - 1)
#define GET_PCID_ADDR(_pcid) (((_pcid) << PHY_CTX_SHIFT) | (_pcid_addr) >> PHY_CTX_SHIFT)

u32 vcid;

vcid = 96;
while (vcid) {
    u32 vcid_addr, pcid_addr, offset;

    vcid--;
```

(Refer to these errata: E4_5706CA0_94 and E1_5706SA0_562)

```

if (CHIP_ID(bp) == CHIP_ID_5706_A0) {
    u32 new_vcid;

    vcid_addr = GET_PCID_ADDR(vcid);
    if (vcid & 0x8) {
        new_vcid = 0x60 + (vcid & 0xf0) + (vcid & 0x7);
    }
    else {
        new_vcid = vcid;
    }
    pcid_addr = GET_PCID_ADDR(new_vcid);
}
else {
    vcid_addr = GET_CID_ADDR(vcid);
    pcid_addr = vcid_addr;
}
vcid_addr = GET_CID_ADDR(vcid);
pcid_addr = vcid_addr;

REG_WR(bp, context.ctx_virt_addr, 0x00);
REG_WR(bp, context.ctx_page_tbl, pcid_addr);

/* Zero out the context. */
for (offset = 0; offset < PHY_CTX_SIZE; offset += 4) {
    CTX_WR(bp, 0x00, offset, 0);
}

REG_WR(bp, context.ctx_virt_addr, vcid_addr);
REG_WR(bp, context.ctx_page_tbl, pcid_addr);
}

```

- 7 Load the NetXtreme II firmware. See “[CPU Initialization](#)” on page 124 for details on how the firmware is loaded.
- 8 (Optional) By default, the device's MAC address is programmed by bootcode after power-on or device reset. If a locally administered address is required, or if the NetXtreme II controller must support more than a single MAC address, it should be programmed by writing the “[EMAC MAC Match Registers \(emac_mac_match\[32\], Offset 0x1410\)](#)” on page 319.

```

val = (mac_addr[0] << 8) | mac_addr[1];
REG_WR(bp, emac.emac_mac_match[0], val);

val = (mac_addr[2] << 24) | (mac_addr[3] << 16) | (mac_addr[4] << 8) | mac_addr[5];
REG_WR(bp, emac.emac_mac_match[1], val);

```



Note: The last two entries of the emac_mac_match register are reserved for firmware use and should not be modified by the driver. See “[Shared Hardware Resources](#)” on page 126 for additional information.

-
- 9 Set the kernel bypass block size to 256 bytes by setting the KNL_BYP_BLK_SIZE field of the “[MQ Configuration Register \(mq_config, Offset 0x3c08\)](#)” on page 405 to 0.

```
val = REG_RD(bp, mq.mq_config);  
val &= ~MQ_CONFIG_KNL_BYP_BLK_SIZE;  
val |= MQ_CONFIG_KNL_BYP_BLK_SIZE_256;  
REG_WR(bp, mq.mq_config, val);
```

- 10 Configure the size of the Kernel Mailbox window by writing 0x50000 to the “[MQ Kernel Windows End Register \(mq_knl_wind_end, Offset 0x3c20\)](#)” on page 409 (enabling a 4-MB window through BAR mapped memory), and set the Kernel Bypass Mailbox window to start immediately after the Kernel Mailbox window by writing 0x50000 to the “[MQ Kernel Bypass Window Start Register \(mq_knl_byp_wind_start, Offset 0x3c1c\)](#)” on page 409.

```
#define MAX_CID_CNT          0x4000  
#define MB_KERNEL_CTX_SHIFT   8  
#define MB_KERNEL_CTX_SIZE    (1 << MB_KERNEL_CTX_SHIFT)  
  
val = 0x10000 + (MAX_CID_CNT * MB_KERNEL_CTX_SIZE);  
REG_WR(bp, mq.mq_knl_wind_end, val);  
REG_WR(bp, mq.mq_knl_byp_wind_start, val);
```

- 11 Configure the chain page size to match the system memory allocation page size by programming the PAGE_SIZE field of the “[RV2P Configuration Register \(rv2p_config, Offset 0x2808\)](#)” on page 383 and the PAGE_SIZE field of the “[TBDR Configuration Register \(tbdr_config, Offset 0x5008\)](#)” on page 453. The following example code programs a page size of 4 KB, which is commonly found on x86 architecture systems.

```
#define BCM_PAGE_BITS 12  
#define BCM_PAGE_SIZE (1 << BCM_PAGE_BITS)  
  
val = (BCM_PAGE_BITS - 8) << 24;  
REG_WR(bp, rv2p.rv2p_config, val);  
val = REG_RD(bp, tbdr.tbdr_config);  
val &= ~TBDR_CONFIG_PAGE_SIZE;  
val |= (BCM_PAGE_BITS - 8) << 24 | 0x40;  
REG_WR(bp, tbdr.tbdr_config, val);
```

- 12 Configure the random back-off seed by writing to the EMAC_BACKOFF_SEED field of the “[EMAC Backoff Seed Register \(emac_backoff_seed, Offset 0x1498\)](#)” on page 319.

```
val = mac_addr[0] + (mac_addr[1] << 8) + (mac_addr[2] << 16) +  
      mac_addr[3] + (mac_addr[4] << 8) + (mac_addr[5] << 16);  
REG_WR(bp, emac.emac_backoff_seed, val);
```

- 13** Program the system message transfer unit (MTU) size by writing to the MTU_SIZE field of the “[EMAC RX MTU Size Register \(emac_rx_mtu_size, Offset 0x149c\)](#)” on page 320. If jumbo frames are supported, the JUMBO_ENA bit should also be set.

```
#define MIN_ETHERNET_PACKET_SIZE60
#define MAX_ETHERNET_PACKET_SIZE1514
#define MAX_ETHERNET_JUMBO_PACKET_SIZE9014
#define ETHERNET_FCS_SIZE4
#define ETHERNET_VLAN_TAG_SIZE4
#define ETH_HLEN 14

val = bp->dev->mtu + ETH_HLEN + ETHERNET_FCS_SIZE;
if (val > (MAX_ETHERNET_PACKET_SIZE + ETHERNET_VLAN_TAG_SIZE)) {
    val |= EMAC_RX_MTU_SIZE_JUMBO_ENA;
}
REG_WR(bp, emac.emac_rx_mtu_size, val);
```

- 14** Enable the Ethernet MAC block to generate interrupts in response to link state changes by setting the LINK bit of the “[EMAC Attention Enable Register \(emac_attention_ena, Offset 0x1408\)](#)” on page 316.

```
REG_WR(bp, emac.emac_attention_ena, EMAC_ATTENTION_ENA_LINK);
```

- 15** Program the host memory address for the status block by programming the “[HC Status Address Low Register \(hc_status_addr_l, Offset 0x6810\)](#)” on page 490 and the “[HC Status Address High Register \(hc_status_addr_h, Offset 0x6814\)](#)” on page 490.

```
REG_WR(bp, hc.hc_status_addr_l,
       (u64) bp->status_blk_mapping & 0xffffffff);
REG_WR(bp, hc.hc_status_addr_h, (u64) bp->status_blk_mapping >> 32);
```

- 16** Program the host memory address for the statistics block by programming the “[HC Statistics Address Low Register \(hc_statistics_addr_l, Offset 0x6818\)](#)” on page 490 and the “[HC Statistics Address High Register \(hc_statistics_addr_h, Offset 0x681c\)](#)” on page 490.

```
REG_WR(bp, hc.hc_statistics_addr_l,
       (u64) bp->stats_blk_mapping & 0xffffffff);
REG_WR(bp, hc.hc_statistics_addr_h, (u64) bp->stats_blk_mapping >> 32);
```

- 17** Program the Host Coalescing registers as suggested in the table below and set the COLLECT_STATS, RX_TMR_MODE, and TX_TMR_MODE bits of the “[HC Configuration Register \(hc_config, Offset 0x6808\)](#)” on page 487.

Table 49: Programming the Host Coalescing Registers

Register	Field	Suggested Value	Comments
hc_tx_quick_cons_trip	VALUE	20	—
	INT	20	—
hc_rx_quick_cons_trip	VALUE	6	—
	INT	6	—
hc_comp_prod_trip	VALUE	0	—
	INT	0	—

Table 49: Programming the Host Coalescing Registers (Cont.)

Register	Field	Suggested Value	Comments
hc_tx_ticks	VALUE	80	—
	INT	80	—
hc_rx_ticks	VALUE	18	—
	INT	18	—
hc_com_ticks	VALUE	0	Not required for L2 only drivers.
	INT	0	Not required for L2 only drivers.
hc_cmd_ticks	VALUE	0	Not required for L2 only drivers.
	INT	0	Not required for L2 only drivers.
hc_stats_ticks	HC_STAT_TICKS	1,000,000	Generate statistics block updates every 1 second.
hc_stat_collect_ticks	HC_STAT_COLL_TICKS	3,000	Internally collect statistics every 3 ms.

[See CQ# TO564]

```

if (CHIP_ID(bp) == CHIP_ID_5706_A1) {
    REG_WR(bp, hc.hc_config, HC_CONFIG_COLLECT_STATS);
}
else {
    REG_WR(bp, hc.hc_config, HC_CONFIG_RX_TMR_MODE |
        HC_CONFIG_TX_TMR_MODE |
        HC_CONFIG_COLLECT_STATS);
}

REG_WR(bp, hc.hc_config, HC_CONFIG_RX_TMR_MODE |
    HC_CONFIG_TX_TMR_MODE | HC_CONFIG_COLLECT_STATS);

```

- 18** Clear the internal statistics counters by setting the CLR_STAT_NOW bit of the “[HC Command Register \(hc_command, Offset 0x6800\)](#)” on page 484.

```
REG_WR(bp, hc.hc_command, HC_COMMAND_CLR_STAT_NOW);
```

- 19** Enable link state changes to generate interrupts by setting the LINK_STATE bit of the “[HC Attention Bits Enable Register \(hc_attn_bits_enable, Offset 0x680c\)](#)” on page 490.

```
REG_WR(bp, hc.hc_attn_bits_enable, STATUS_ATTN_BITS_LINK_STATE);
```

- 20** Initialize the appropriate receive filtering features by writing to the “[EMAC RX Mode Register \(emac_rx_mode, Offset 0x14c8\)](#)” on page 327 and the “[RPM Sort User 0 Register \(rpm_sort_user0, Offset 0x1820\)](#)” on page 345. Features that may be enabled/disabled through these registers include Promiscuous mode, broadcast/multicast address filtering, VLAN tag filtering, VLAN tag stripping, perfect match address filtering, and flow control frame processing.

```

rx_mode = SORT_MODE;
sort_mode = MC_EN | BC_EN | 1;
REG_WR(bp, emac.emac_rx_mode, rx_mode);
REG_WR(bp, rpm.rpm_sort_user0, 0x0);
REG_WR(bp, rpm.rpm_sort_user0, sort_mode);

```

```
REG_WR(bp, rpm.rpm_sort_user0, sort_mode | ENA);
```

- 21** Synchronize with the firmware. See “[Driver/Firmware Mailboxes](#)” on page 136 of [Section 8: “Driver/Firmware Shared Memory”](#) for details on driver/firmware synchronization. Following a controller reset, this synchronization allows the bootcode firmware to perform any additional initialization or implement any applicable errata fixes that may be necessary before the controller is fully enabled to send and receive Ethernet traffic.

```
fw_sync(bp, DRV_MSG_DATA_WAIT2 | DRV_MSG_CODE_RESET);
```

- 22** Enable the remainder of the NetXtreme II blocks and delay for 20 µs to allow the various blocks to complete initialization. The NetXtreme II controller is now ready to send and receive traffic.

```
REG_WR(bp, misc.misc_enable_set_bits, 0xffffffff);
REG_RD(bp, misc.misc_enable_set_bits);
udelay(20);
```

CPU INITIALIZATION

The NetXtreme II controller operates through a combination of hardware and firmware. The firmware must be loaded by the device driver; otherwise, the controller is not able to send or receive traffic. The firmware is provided by Broadcom as a group of C header files that may be included in the driver. The following procedures must be used to load the provided firmware.

RV2P FIRMWARE LOAD PROCEDURE AND PSEUDO CODE

The RV2P processors use a different file format and load procedure than the other integrated processors (such as the receive processor or RXP). The RV2P processor code is included in C header files named rv2p_proc1.h and rv2p_proc2.h, which contain an array of 64-bit instruction words that are written to the RV2P processor through the “[RV2P Instruction High Register \(rv2p_instr_high, Offset 0x2830\)](#)” on page 384 and the “[RV2P Instruction Low Register \(rv2p_instr_low, Offset 0x2834\)](#)” on page 385. After the instruction words have been written to the instruction registers, an additional address/command register (see “[RV2P Processor 1 Address/Command Register \(rv2p_proc1_addr_cmd, Offset 0x2838\)](#)” on page 386 and “[RV2P Processor 2 Address/Command Register \(rv2p_proc2_addr_cmd, Offset 0x283c\)](#)” on page 386) must be written to write the instruction into the RV2P instruction RAM. After all of the instructions have been written, the RV2P processor must be reset by writing to the “[RV2P Command Register \(rv2p_command, Offset 0x2800\)](#)” on page 382.

```
#define RV2P_PROC1          0
#define RV2P_PROC2          1

load_rv2p_fw(struct bnx2_dev *bp, u32 *rv2p_code, u32 rv2p_code_len,
             u32 rv2p_proc)
{
    int i;
    u32 val;

    /* Read through the instruction array for the RV2P processor */
    for (i = 0; i < rv2p_code_len; i += 8) {
        REG_WR(bp, rv2p.rv2p_instr_high, *rv2p_code);
        rv2p_code++;
        REG_WR(bp, rv2p.rv2p_instr_low, *rv2p_code);
        rv2p_code++;
    }

    /* Write the instruction to the appropriate instruction */
    /* RAM address. */
    if (rv2p_proc == RV2P_PROC1) {
        val = (i / 8) | RV2P_PROC1_ADDR_CMD_RDWR;
        REG_WR(bp, rv2p.rv2p_proc1_addr_cmd, val);
    }
    else {
        val = (i / 8) | RV2P_PROC2_ADDR_CMD_RDWR;
        REG_WR(bp, rv2p.rv2p_proc2_addr_cmd, val);
    }
}

/* Reset the processor */
if (rv2p_proc == RV2P_PROC1) {
    REG_WR(bp, rv2p.rv2p_command, RV2P_COMMAND_PROC1_RESET);
}
else {
    REG_WR(bp, rv2p.rv2p_command, RV2P_COMMAND_PROC2_RESET);
}
```

SHARED HARDWARE RESOURCES

Because the NetXtreme II controller relies on a combination of host device driver and internal firmware to operate, certain hardware resources are reserved for firmware use and should not be used by the host device driver. These hardware resources include the following:

- Entries 28 to 31 of the “[EMAC MAC Match Registers \(emac_mac_match\[32\], Offset 0x1410\)](#)” on page 319, comprising the last two perfect MAC match filters
- Rules 10 through 15 of the RPM Rule Check Control XX registers (see “[RPM Rule Check Control 10 Register \(rpm_rc_ctrl_10, Offset 0x1950\)](#)” on page 357) and the RPM Rule Check Value/Mask XX registers (see “[RPM Rule Check Value/Mask 10 Register \(rpm_rc_value_mask_10, Offset 0x1954\)](#)” on page 357)
- The “[RPM VLAN Match 3 Register \(rpm_vlan_match3, Offset 0x181c\)](#)” on page 344
- The “[RPM Sort User 3 Register \(rpm_sort_user3, Offset 0x182c\)](#)” on page 346
- GPIO pins (and their corresponding registers)
 - 0,1, and 2 for NIC implementations
 - 6 and 7 for LOM implementations that use UMP firmware



Section 6: PCI

EXPANSION ROM

DESCRIPTION

The expansion ROM on the Broadcom NetXtreme II family is intended for implementation of Preboot Execution Environment (PXE) and iSCSI boot. The Broadcom NetXtreme II family supports up to 64-KB Expansion ROM.

OPERATIONAL CHARACTERISTICS

By default, the Expansion ROM is disabled and the firmware must explicitly enable this feature by setting the EXP_ROM_SIZE field of the PCI Configuration 2 Register (see “[PCI Configuration 2 Register \(pci_config_2, Offset 0x408\)](#)” on page 204).

BIOS

The BIOS detects whether a PCI device supports an Expansion ROM by writing the value 0xFFFFFFFF to the Expansion ROM Base Address register. If the value is non-0 when the BIOS reads back from this register, this PCI device supports Expansion ROM—otherwise it is not supported. If the EXP_ROM_SIZE field of the PCI Configuration 2 register (see “[PCI Configuration 2 Register \(pci_config_2, Offset 0x408\)](#)” on page 204) is set to a value of 0 in the case of the Broadcom NetXtreme II, then Broadcom NetXtreme II returns a value of 0x00000000 when the PCI Expansion ROM Base Address register is read, indicating that Expansion ROM is not supported. Otherwise, it returns a non-0 value that indicates the size of the expansion ROM supported by the NetXtreme II. (Refer to [PCI] for details.)

If a PCI device supports Expansion ROM, the BIOS will assign a Expansion Base address to the device. It then checks for a valid ROM header (0x55 0xAA as first 2 bytes, and so on) and checksum. If the ROM header and image are valid, the BIOS will copy the Expansion ROM image to HOST's Upper Memory Block (UMB) and invoke the initializing entry point.

Accesses to the PCI Expansion ROM are handled by firmware running on the NetXtreme II (generally referred to as the bootcode). When the host generates a PCI access to the Expansion ROM, the address is latched in the PCI Expansion ROM Address register (see “[PCI Expansion ROM Address Register \(pci_exp_rom_addr, Offset 0x420\)](#)” on page 210) and an EXP_ROM_ATTN is asserted. When the firmware services this attention, it places the expansion ROM data read from NVRAM into the PCI Expansion ROM Data register (see “[PCI Expansion ROM Address Register \(pci_exp_rom_addr, Offset 0x420\)](#)” on page 210) which places the data onto the PCI bus and completes the Expansion ROM access. Between the time that the EXP_ROM_ATTN is asserted and the final data is written to the PCI Expansion ROM Data register, the PCI block issues retries on the PCI bus to make the requester wait.

BYTE-SWAPPING ISSUES

BACKGROUND

There are two common formats for storing data in memory, big-endian architecture and little-endian architecture. The endianess of a system is determined by the method used to store multi-byte quantities in memory. A big-endian system stores the most significant byte (MSB) of a multi-byte quantity at the lowest memory address, while a little-endian system stores the least significant byte (LSB) of a multi-byte quantity at the lowest memory address. The following example demonstrates how a series of single-byte and multi-byte values are interpreted differently on big-endian versus little-endian systems.

Memory Address	Memory Contents
0x100	0x80
0x101	0x81
0x102	0x82
0x103	0x83
0x104	0x84
0x105	0x85
0x106	0x86
0x107	0x87

The 8-byte value at address 0x100 is 0x8081828384858687 on a big-endian system, and 0x8786858483828180 on a little-endian system

The byte at address 0x100 is 0x80 on both big-endian and little-endian systems

The 2-byte value at address 0x102 is 0x8283 on a big-endian system, and 0x8382 on a little-endian system

The 4-byte value at address 0x104 is 0x84858687 on a big-endian system, and 0x87868584 on a little-endian system

Figure 17: Big-Endian Versus Little-Endian Data Representation

The Intel® Pentium® 4 processor and PCI bus are examples of little-endian architecture while the IBM® PowerPC® processor is an example of big-endian architecture.

ARCHITECTURE

The Broadcom NetXtreme II controller operates internally using a 64-bit big-endian architecture, and its internal RISC processors are 64-bit big-endian devices. This differs from many host systems that connect to the NetXtreme II through the little-endian PCI/PCI-X/PCIe bus, and operate with little-endian processors. To accommodate the difference in data representation between the internal and external interfaces, the Broadcom NetXtreme II controller provides several different byte-swapping options so that both big-endian and little-endian hosts can interface seamlessly over the PCI interface. The following registers are provided to detect and adjust the byte-swapping configuration of the NetXtreme II controller:

- The swap diagnostics registers (see [“PCI Swap Diagnostics 0 Register \(pci_swap_diag0, Offset 0x418\)” on page 209](#) and [“PCI Swap Diagnostics 1 Register \(pci_swap_diag1, Offset 0x41c\)” on page 209](#)) are used to read fixed values from register space and detect swapping errors.
- The TARGET_BYTE_SWAP bit of the [“Miscellaneous Configuration Register \(pcicfg_misc_config, Offset 0x68\)” on page 172](#) controls whether byte-swapping is enabled on all mailbox registers, memory-mapped I/O registers, and indirect register accesses.
- The TARGET_MB_WORD_SWAP bit of the [“Miscellaneous Configuration Register \(pcicfg_misc_config, Offset 0x68\)” on page 172](#) controls whether 32-bit words are swapped when accessing the NetXtreme II mailbox registers through memory-mapped I/O (i.e. through memory space mapped by the PCI BAR_1 register).
- The TARGET_GRC_WORD_SWAP bit of the [“Miscellaneous Configuration Register \(pcicfg_misc_config, Offset 0x68\)” on page 172](#) controls whether 32-bit words are swapped when accessing NetXtreme II registers through memory-mapped I/O (i.e. through memory space mapped by the PCI BAR_1 register) or through indirect access (using the [“Register Window Address Register \(pcicfg_reg_window_address, Offset 0x78\)” on page 176](#) and the [“Register Window Data Register \(pcicfg_reg_window, Offset 0x80\)” on page 176](#)).
- The DATA_BYTE_SWAP bit of the [“DMA Configuration Register \(dma_config, Offset 0xc08\)” on page 281](#) controls whether byte-swapping is enabled for all DMA transactions on frame data.
- The DATA_WORD_SWAP bit of the [“DMA Configuration Register \(dma_config, Offset 0xc08\)” on page 281](#) controls whether 32-bit words are swapped for all DMA transactions on frame data.
- The CNTL_BYTE_SWAP bit of the [“DMA Configuration Register \(dma_config, Offset 0xc08\)” on page 281](#) controls whether byte-swapping is enabled for all DMA transactions on control data.
- The CNTL_WORD_SWAP bit of the [“DMA Configuration Register \(dma_config, Offset 0xc08\)” on page 281](#) controls whether 32-bit words are swapped for all DMA transactions on control data.

BYTE-SWAPPING VERSUS WORD-SWAPPING

The NetXtreme II controller provides controls for both byte-swapping and word-swapping. The difference between these two swapping modes can be seen in [Figure 18](#).

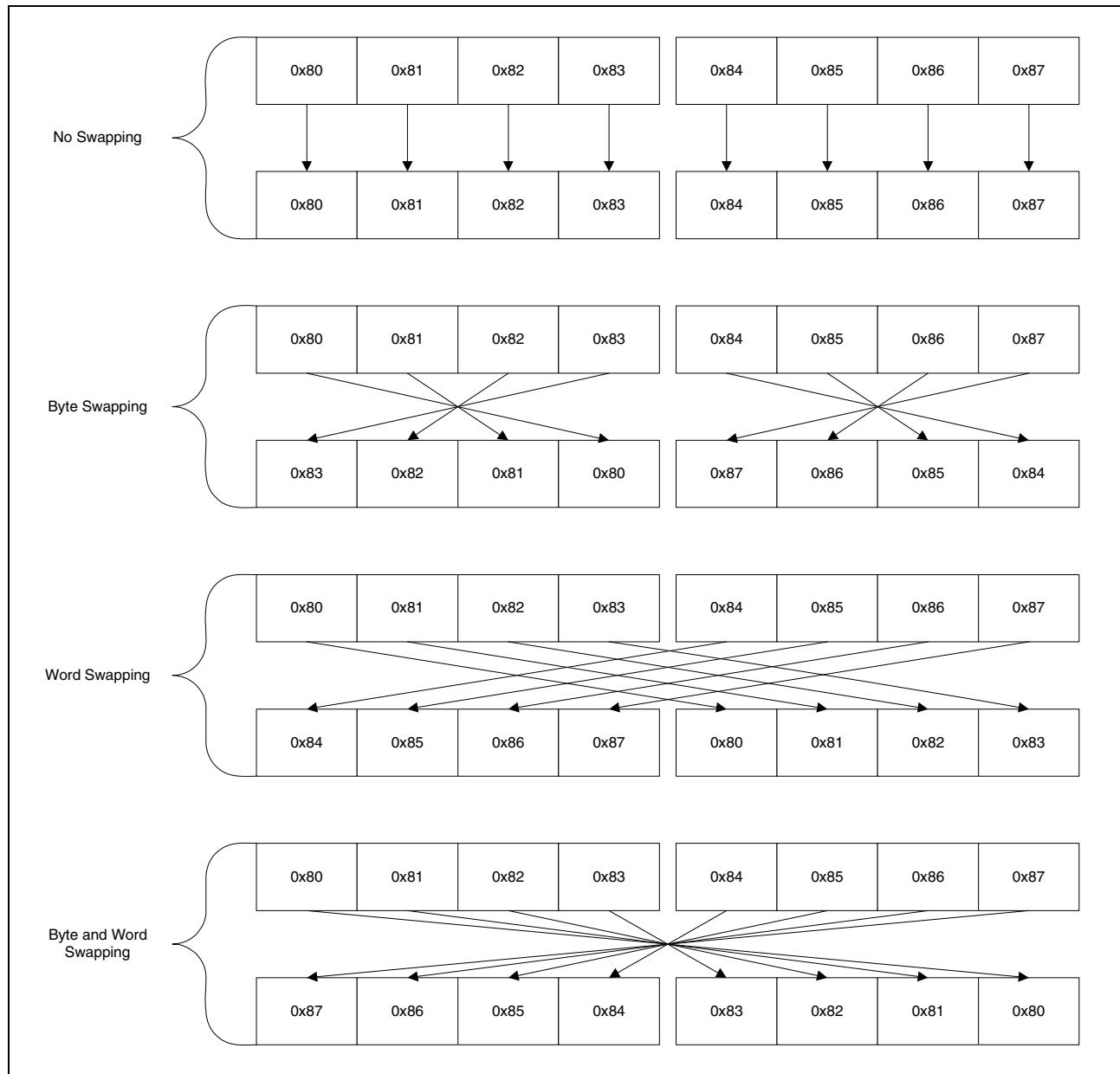


Figure 18: Byte-Swapping Versus Word-Swapping

Byte-swapping modifies individual bytes within a 32-bit word, while word-swapping modifies entire 32-bit words within a 64-bit value.

SELECTING THE CORRECT SWAPPING MODES

The Broadcom NetXtreme II controller provides individual byte-swapping and word-swapping capabilities for register access, mailbox access, and DMA control/data transfers.

ENABLING SWAPPING FOR PCI TARGET ACCESSES

The first phase in selecting the correct swapping mode involves setting the correct swapping mode for register (or GRC) accesses. (If register accesses are not working correctly, then the NetXtreme II controller cannot be configured for any other operations.) The following example shows the effects of different combinations of the TARGET_BYTE_SWAP and TARGET_GRC_WORD_SWAP bits of the “[Miscellaneous Configuration Register \(pcicfg_misc_config, Offset 0x68\)](#)” on page 172 when accessing the swap diagnostics registers (see “[PCI Swap Diagnostics 0 Register \(pci_swap_diag0, Offset 0x418\)](#)” on page 209 and “[PCI Swap Diagnostics 1 Register \(pci_swap_diag1, Offset 0x41c\)](#)” on page 209). This example was performed on a BCM5706 controller installed in a 64-bit PCI-X slot on an x86 processor.

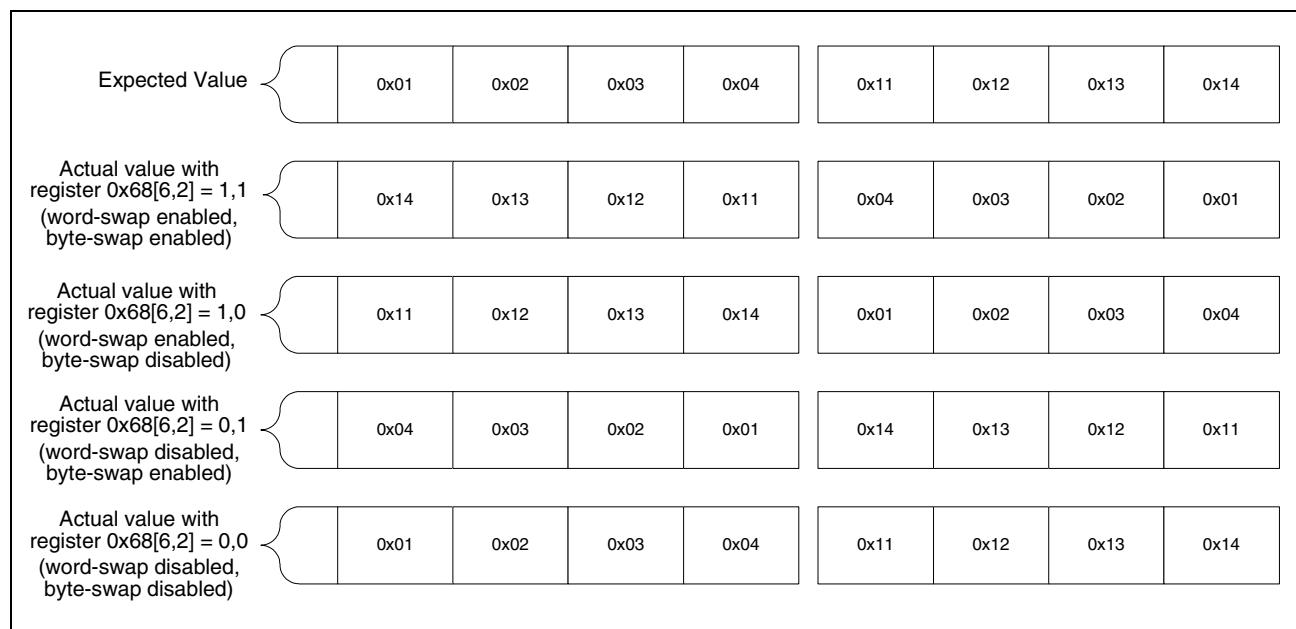


Figure 19: Byte-Swapping and Word-Swapping Example

The example in [Figure 19](#) shows that the correct byte-swapping and word-swapping settings for register (or GRC) accesses on the little-endian x86 architecture are TARGET_BYTE_SWAP = 0 and TARGET_GRC_WORD_SWAP = 0. In some cases, the operating system may provide support for handling the conversion between big-endian and little-endian values, and is generally implemented as an operation performed by the host CPU (such as Linux®). If this is the case, then byte-swapping and word-swapping should not be enabled on the NetXtreme II controller for PCI target accesses.

ENABLING SWAPPING FOR DMA ACCESSES

The NetXtreme II controller performs DMA reads and writes for both control and frame data. The control data consists of data structures defined internally by the NetXtreme II controller that are read from or written to host memory through DMA operations, and include structures such as the “[Host Status Block \(Status\)](#)” on page 73, “[Host Statistics Block \(Stat\)](#)” on page 81, and “[Receive Buffer Descriptor Chain Record \(rx_bd\)](#)” on page 90. The frame data refers to the actual Ethernet frames that are sent to or received from the MAC interface.

Unlike the previous PCI target accesses, control data must be correctly swapped by the NetXtreme II controller to allow the host operating system device driver to read the various data structures correctly. The host CPU cannot perform this operation since it is not involved in the data transfer. Also unlike the previous PCI target configuration, there are no registers than can be used to identify the correct swap settings programmatically, so the device driver developer must select the correct settings for the system architecture in use. For little-endian systems such as those that use an x86 processor, the device driver should set the CNTL_WORD_SWAP bit of the “[DMA Configuration Register \(dma_config, Offset 0xc08\)](#)” on page 281 for proper operation. For big-endian systems the device driver should also set the CNTL_BYTE_SWAP bit of the same register.

Byte-swapping and word-swapping should always be enabled for the DATA_BYTE_SWAP and DATA_WORD_SWAP bits of the “[DMA Configuration Register \(dma_config, Offset 0xc08\)](#)” on page 281, no matter what endian architecture is used by system. Ethernet frames are always interpreted in big-endian format, so setting the bits DATA_BYTE_SWAP =1 and DATA_WORD_SWAP = 1 always converts the little-endian PCI bus transactions back to big-endian format for use by the operating system upper layer protocols.

The only exception to the previous settings for control and frame data apply to system architectures that attempt to perform little-endian to big-endian conversions at the PCI bridge level, without any involvement by the NetXtreme II controller or the host CPU. In these cases, the device driver developer must consult with the PCI bridge documentation to select the proper byte-swap and word-swap settings for the NetXtreme II controller.

Section 7: Interrupt Processing

INTERRUPT PROCESSING

The NetXtreme II family of controllers supports an Interrupt Coalescing or Interrupt Moderation mechanism to reduce the overhead of interrupt processing under heavy loads. The NetXtreme II also supports both PCI line interrupts and MSI.

HOST COALESCING

The Host Coalescing (HC) engine is responsible for pacing the rate at which the NetXtreme II updates the Host Status Block and Host Statistics Block. After an update of either block occurs, the HC engine will generate an interrupt (either a PCI line interrupt or an MSI).

By default, the NetXtreme II family generates a status block update every time one of the fields in the Host Status Block is updated. This behavior can be modified by updating the trip point registers listed below. A trip point can occur after a specified number of updates have occurred or after a specified time interval has elapsed from a previous update. Although each trip point is evaluated separately, when one trip point is reached, all updates will occur at once and all timers/counters will be reset.

A host update occurs whenever one of the following criteria is met:

- The number of TX Quick BD Chain entries is greater than or equal to the value set in the HC TX Quick Consumer Trip Point register (see “[HC TX Quick Consumer Trip Pointer Register \(hc_tx_quick_cons_trip, Offset 0x6820\)](#)” on page 490).
- The TX coalescing timer is greater than or equal to the value set in the HC TX Ticks register (see “[HC TX Ticks Register \(hc_tx_ticks, Offset 0x6830\)](#)” on page 492) and new TX Quick BD Chain entries have been added.
- The number of RX Quick BD Chain entries is greater than or equal to the value set in the HC RX Quick Consumer Trip Point register (see “[HC RX Quick Consumer Trip Point Register \(hc_rx_quick_cons_trip, Offset 0x6828\)](#)” on page 491).
- The RX coalescing timer is greater than or equal to the value set in the HC RX Ticks register (see “[HC RX Ticks Register \(hc_rx_ticks, Offset 0x682c\)](#)” on page 492) and new RX Quick BD Chain entries have been added.
- The number of Kernel Completion Queue entries is greater than or equal to the value set in the HC Kernel Completion Producer Trip Point register (see “[HC Kernel Completion Producer Trip Point Register \(hc_comp_prod_trip, Offset 0x6824\)](#)” on page 491).
- The Kernel Completion Queue coalescing timer is greater than or equal to the value set in the HC Kernel Completion Ticks register (see “[HC Kernel Completion Ticks Register \(hc_com_ticks, Offset 0x6834\)](#)” on page 493) and new Kernel Completion Queue entries have been added.
- The Kernel Command Queue timer is greater than or equal to the value set in the HC Command Ticks register (see “[HC Command Ticks Register \(hc_cmd_ticks, Offset 0x6838\)](#)” on page 493) and new Kernel Command Queue entries have been added.
- The HC Periodic Ticks Timer is greater than or equal to the value set in the HC Periodic Ticks register (see “[HC Periodic Ticks Register \(hc_periodic_ticks, Offset 0x683c\)](#)” on page 494) and any indexes maintained by the Host Status Block have changed since the last update.

Furthermore, several of the above registers allow different values to be set when the host driver is executing its Interrupt Service Routine (ISR).

MESSAGE SIGNALLED INTERRUPTS

The PCI 2.2 specification defined a new mechanism for a device to request services by its device driver called Message Signaled Interrupts (MSI). In MSI, a PCI device will DMA a particular message to a specified address in host memory if it needs to request services by its device driver. This host memory write will be translated by the system to an interrupt and forwarded to the host CPU for further processing. The main advantages of MSI generation versus the traditional PCI line interrupt mechanism eliminates the need:

- For an interrupt signal trace to the PCI device.
- To perform a dummy read from the device by the device driver in its ISR. This is done to force all posted writes to be flushed to the host memory.
- For multiple devices to share a single interrupt line.

ATTENTIONS

The NetXtreme II family uses an internal Attention mechanism to signal important controller events such as link state change or parity error detection. These internal events can be enabled to update the Host Status Block and generate interrupts to the host operating system. See [“HC Attention Bits Enable Register \(hc_attn_bits_enable, Offset 0x680c\)” on page 490](#) and the [“Host Status Block \(Status\)” on page 73](#) for a complete description of attentions that may be enabled. At a minimum it is expected that a device driver will enable the LINK_STATE attention.

INTERRUPT PROCEDURE

The following steps highlight the Broadcom-recommended actions for the NetXtreme II ISR.

- 1 Verify that an interrupt was asserted. Read the Index field of the Status Block (see [“Host Status Block \(Status\)” on page 73](#)) and verify the current value of that field against the last value seen. If the values are equal then exit the ISR.
- 2 Acknowledge the interrupt. Set the MASK_INT and USE_INT_HC_PARAM fields of the Interrupt Acknowledge Command Register (see [“Interrupt Acknowledge Command Register \(pcicfg_int_ack_cmd, Offset 0x84\)” on page 177](#)). This masks additional interrupts and force the Host Coalescing (HC) engine to use the *In Interrupt* coalescing parameters.
- 3 Save the current Index field from the Host Status Block. The Index field is used as a sort of sequence number that allows the ISR to tell the NetXtreme II which Host Status Block update was last processed.
- 4 Check for internal attentions. Read the Host Status Block Attention Bits field (see [“Host Status Block Attention Bits \(status_attn_bits\)” on page 75](#)) of the Host Status Block and compare the value to the Host Status Block Attention Acknowledge field (see [“Host Status Block Attention Acknowledge \(status_attn_bits_ack\)” on page 77](#)). If the fields are not equal then an internal attention has been generated. See [“Host Status Block Attention Bits \(status_attn_bits\)” on page 75](#) for the correct action to take when an attention is asserted.
- 5 Check for RX traffic. Compare the value of the RX Quick Consumer Index field (see [“RX Quick Consumer Index 0 \(status_rx_quick_consumer_index0\)” on page 77](#)) of the Host Status Block to the driver copy of the field from the received packet. If the values are not equal then a packet has been received.
- 6 Check for TX traffic completes. Compare the value of the TX Quick Consumer Index field (see [“TX Quick Consumer Index 0 \(status_tx_quick_consumer_index0\)” on page 77](#)) of the Host Status Block to the driver copy of the field from the last sent packet. If the values are not equal then a packet has been received.
- 7 Check for additional status block updates. Compare the current value of the Index field from the Host Status Block to the value saved in step 3 above. If the values are not equal, proceed to step 3.
- 8 Enable interrupts. Write the most recently saved Index value from step 3 to the INDEX field and clear the MASK_INT and USE_INT_HC_PARAM fields of the Interrupt Acknowledge Command register (see [“Interrupt Acknowledge Command Register \(pcicfg_int_ack_cmd, Offset 0x84\)” on page 177](#)).



Section 8: Driver/Firmware Shared Memory

The section is drawn from the shmem.h file of the BCM5706 bootcode v1.0.2, dated 2/23/2005.

The Broadcom NetXtreme II supports a shared memory interface that allows the host driver to communicate with the firmware running on the embedded RISC processors. Depending on the feature set provided by the driver, some or all of these shared memory mailboxes are accessed by the driver. This section provides details on that interface.

The shared memory area is implemented in the scratchpad memory area of the management processor (see [“MCP Scratchpad Area \(mcp_scratch\[8192\], Offset 0x160000\)” on page 563](#). Legacy versions of the NetXtreme II bootcode implemented the shared memory interface at a fixed offset from the beginning of the scratchpad memory, but current versions of bootcode use a magic value at a fixed offset followed by a pointer to the actual location in shared memory. The following code demonstrates the proper method to find the shared memory interface:

```
#define MCP_SCRATCH          0x00160000
#define HOST_VIEW_SHMEM_BASE   0x00167c00

#define SHM_HDR_SIGNATURE      MCP_SCRATCH
#define SHM_HDR_SIGNATURE_SIG_MASK 0xfffff0000
#define SHM_HDR_SIGNATURE_SIG    0x53530000
#define SHM_HDR_SIGNATURE_VER_MASK 0x000000ff
#define SHM_HDR_SIGNATURE_VER_ONE 0x00000001

#define SHM_HDR_ADDR_0          MCP_SCRATCH + 4

val = REG_RD_IND(bp, SHM_HDR_SIGNATURE);

if ((val & SHM_HDR_SIGNATURE_SIG_MASK) == SHM_HDR_SIGNATURE_SIG)
    bp->shmem_base = REG_RD_IND(bp, SHM_HDR_ADDR_0);
else
    bp->shmem_base = HOST_VIEW_SHMEM_BASE;
```

Because the management processor of the NetXtreme II is a big-endian processor, all of the following shared memory locations are documented in the native big-endian format, unless specified otherwise. The shared memory area has the following structure:

```
/* Total size should be exactly 1k bytes */
typedef struct _shmem_region_t
{
    drv_fw_mb_t           drv_fw_mb;        /* 0x000 - 0x01f */
    dev_info_t             dev_info;         /* 0x020 - 0x1bf */
    bc_state_t              bc_state;        /* 0x1c0 - 0x1df */
    u32_t reserved[0x34];     license_key_t      fw_lic_key;       /* 0x1e0 - 0x213
*/
    mgmtfw_state_t        mgmtfw_state;     /* 0x214 - 0x353 */
    u32_t reserved[120];
    license_key_t          drv_lic_key;      /* 0x3cc - 0x3ff */
} shmem_region_t;
```

DRIVER/FIRMWARE MAILBOXES

The driver/firmware mailboxes provide a mechanism for the host operating system driver to communicate with the firmware to coordinate their operation. The following C data structure shows the layout of the driver/firmware mailbox:

```

typedef struct _drv_fw_mb_t
{
    u32_t drv_reset_signature;
        #define DRV_RESET_SIGNATURE          0x4841564b /* HAVK */
        #define BIOS_SIGNATURE               0x534f4942

        #define HWKEY_SKIP_MAC_TIMEOUT_US   10000000

    u32_t drv_mb;
        #define DRV_MSG_CODE                0xffff000000
        #define DRV_MSG_CODE_RESET          0x01000000
        #define DRV_MSG_CODE_UNLOAD         0x02000000
        #define DRV_MSG_CODE_SHUTDOWN       0x03000000
        #define DRV_MSG_CODE_SUSPEND_WOL   0x04000000
        #define DRV_MSG_CODE_FW_TIMEOUT    0x05000000
        #define DRV_MSG_CODE_UNUSED         0x06000000
        #define DRV_MSG_CODE_DIAG           0x07000000
        #define DRV_MSG_CODE_VALIDATE_KEY   0x08000000
        #define DRV_MSG_CODE_SUSPEND_NO_WOL 0x09000000
        #define DRV_MSG_CODE_GET_CURR_KEY   0xa0000000
        #define BIOS_MSG_CODE_HANDSHAKE     0xff000000

        #define DRV_MSG_DATA                0x00ff0000
        #define DRV_MSG_DATA_WAIT0          0x00010000
        #define DRV_MSG_DATA_WAIT1          0x00020000
        #define DRV_MSG_DATA_WAIT2          0x00030000
        #define DRV_MSG_DATA_WAIT3          0x00040000
        #define DRV_MSG_DATA_WAIT_RESET    0x00050000
        #define DRV_MSG_DATA_WAIT4          0x00060000
        #define BIOS_MSG_DATA_REQ           0x00010000
        #define BIOS_MSG_DATA_CONFIRM        0x00020000

        #define DRV_MSG_SEQ                 0x0000ffff

        #define FW_ACK_TIME_OUT_MS          50

    u32_t fw_mb;
        #define FW_MSG_ACK                 0x0000ffff
        #define FW_MSG_STATUS_MASK          0x00ff0000
        #define FW_MSG_STATUS_OK            0x00000000
        #define FW_MSG_STATUS_FAILURE       0x00ff0000
        #define FW_SIGN_PRESERVE_MEMORY    0x55aa5a5a

    u32_t link_status;
        #define NETLINK_GET_LINK_STATUS_INIT_VALUE 0xffffffff
        #define NETLINK_GET_LINK_STATUS_LINK_UP      0x1
        #define NETLINK_GET_LINK_STATUS_LINK_DOWN    0x0
        #define NETLINK_GET_LINK_STATUS_SPEED_MASK   0x1e
        #define NETLINK_GET_LINK_STATUS_AN_INCOMPLETE (0<<1)

```

```

#define NETLINK_GET_LINK_STATUS_10HALF          (1<<1)
#define NETLINK_GET_LINK_STATUS_10FULL           (2<<1)
#define NETLINK_GET_LINK_STATUS_100HALF          (3<<1)
#define NETLINK_GET_LINK_STATUS_100BASE_T4        (4<<1)
#define NETLINK_GET_LINK_STATUS_100FULL           (5<<1)
#define NETLINK_GET_LINK_STATUS_1000HALF          (6<<1)
#define NETLINK_GET_LINK_STATUS_1000FULL          (7<<1)
#define NETLINK_GET_LINK_STATUS_2500HALF          (8<<1)
#define NETLINK_GET_LINK_STATUS_2500FULL          (9<<1)
#define NETLINK_GET_LINK_STATUS_AN_ENABLED        (1<<5)
#define NETLINK_GET_LINK_STATUS_AN_COMPLETE       (1<<6)
#define NETLINK_GET_LINK_STATUS_PARALLEL_DET     (1<<7)
#define NETLINK_GET_LINK_STATUS_RESERVED          (1<<8)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_1000FULL (1<<9)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_1000HALF (1<<10)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_100BT4   (1<<11)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_100FULL  (1<<12)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_100HALF  (1<<13)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_10FULL   (1<<14)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_10HALF   (1<<15)
#define NETLINK_GET_LINK_STATUS_TX_FC_ENABLED      (1<<16)
#define NETLINK_GET_LINK_STATUS_RX_FC_ENABLED      (1<<17)
#define NETLINK_GET_LINK_STATUS_PARTNER_SYM_PAUSE_CAP (1<<18)
#define NETLINK_GET_LINK_STATUS_PARTNER_ASYM_PAUSE_CAP (1<<19)
#define NETLINK_GET_LINK_STATUS_SERDES_LINK        (1<<20)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_2500FULL (1<<21)
#define NETLINK_GET_LINK_STATUS_PARTNER_AD_2500HALF (1<<22)
#define DRV_PULSE_PERIOD_MS                      250

u32_t drv_pulse_mb;
#define DRV_PULSE_SEQ_MASK                     0x00007fff
#define DRV_PULSE_SYSTEM_TIME_MASK             0xfffff000
#define DRV_PULSE_ALWAYS_ALIVE                0x00008000

u32_t reserved[3];
} drv_fw_mb_t;

```

DRIVER RESET SIGNATURE MAILBOX

The driver reset signature mailbox must be written by the driver before the NetXtreme II controller is reset. This allows the bootcode to know that the reset was caused by software and modifies the bootcode initialization routines to account for this situation. After the DRV_RESET_SIGNATURE value has been written, the driver must then update the “[Driver Mailbox \(drv_mb\)](#)”.

DRIVER MAILBOX (DRV_MB)

The driver mailbox is used by the operating system driver to pass status information and commands to the firmware. It is implemented as a 32-bit word that includes an 8-bit command code, an 8-bit data element, and a 16-bit sequence number. The sequence number should begin at 0 and increment by 1 for each command/data pair that is sent to the firmware, rolling over to 0 as necessary.

The most common command used by a device driver is the RESET command, which signals the firmware that the driver is about to reset the NetXtreme II hardware. A driver issues the RESET command along with the WAIT0 data and waits for the



firmware to respond (see “[Firmware Mailbox \(fw_mb\)](#)”). If the firmware does not acknowledge the command within the timeout period (50 ms), the driver should write the FW_TIMEOUT command (without updating the sequence number) to the driver mailbox and proceed to reset the controller.

After a controller reset occurs, the driver must wait for the firmware to load and complete its phase one initialization before continuing. The driver uses the RESET command again, along with the WAIT1 data, before continuing. If the firmware does not acknowledge the command within the timeout period (50 ms), the driver should write the FW_TIMEOUT command (without updating the sequence number) and continue its initialization. It is also recommended that the driver generate an error message or log an event in the system error log if this situation occurs.

DRIVER PULSE MAILBOX (DRV_PULSE_MB)

The drv_pulse_mb mailbox contains a monotonically increasing value (between 0x0 and 0x7ff) that is regularly written by the driver to indicate that the driver is still active. The driver should update this mailbox every 250 ms, along with the current system time. (The current system time is used in conjunction with the license key to enforce license expiration.) If the bootcode sees that the driver is not active (i.e., it has not updated the drv_pulse_mb in over 3 seconds), bootcode updates its internal state to indicate that the driver is not present and invalidates the license key.

The driver pulse mailbox also contains a field to represent the system time, which should be updated on a periodic basis by the driver. This field is used to enforce the key expiration field of the license key and is specified by the following equation:

$$(Year - 2001) * 12 * 32 + month * 32 + day$$

FIRMWARE MAILBOX (FW_MB)

The firmware mailbox is used by the NetXtreme II bootcode to post messages back to the driver in response to a command received from the “[Driver Mailbox \(drv_mb\)](#)”. The only message currently supported is an acknowledgement (ACK) to a previously posted command. Whenever the bootcode posts the FW_MSG_ACK, the driver is assured that the command was received and processed.

FIRMWARE SYNCHRONIZATION PSEUDO CODE

The driver should use the following code to implement message passing using the driver/firmware mailboxes discussed above:

```
static int fw_sync(struct dev *bp, u32 msg_data)
{
    int i;
    u32 val;

    /* Skip further processing if a firmware timeout */
    /* has already occurred. */
    if (bp->fw_timed_out)
        return ERROR;

    /* Increment the driver maintained sequence number. */
    bp->fw_wr_seq++;
    msg_data |= bp->fw_wr_seq;

    /* Write the message, data, and sequence number to the */
    /* driver mailbox. */
    SHMEM_WR(drv_mb, msg_data);
```



```

/* Wait for an acknowledgement. */
for (i = 0; i < (FW_ACK_TIME_OUT_MS * 1000)/5; i++) {
    /* Delay for 5us. */
    udelay(5);

    val = SHMEM_RD(fw_mb);

    /* Exit the loop if an acknowledgement is received. */
    if ((val & FW_MSG_ACK) == (msg_data & DRV_MSG_SEQ))
        break;
}

/* Let the firmware know that a timeout occurred. */
if (((val & FW_MSG_ACK) != (msg_data & DRV_MSG_SEQ)) &&
    ((msg_data & DRV_MSG_DATA) != DRV_MSG_DATA_WAIT0)) {

    msg_data &= ~DRV_MSG_CODE;
    msg_data |= DRV_MSG_CODE_FW_TIMEOUT;

    SHMEM_WR(drv_mb, msg_data);

    bp->fw_timed_out = 1;

    return ERROR;
}

return SUCCESS;
}

```

DEVICE INFORMATION (DEV_INFO)

The device information area allows the driver to access the NVRAM configured options of the NetXtreme II controller without being forced to use the slow NVRAM access mechanisms documented in [Section 3: “NVRAM Configuration” on page 35](#). The details of this structure are further broken down as follows:

- Shared hardware configuration—This structure describes hardware features that are common to all elements of the controller. Examples of this include the settings and usage of GPIO pins.
- Port-specific configuration—This structure describes hardware features that are specific to a particular port of the controller. Examples of this include the MAC address of the port.

Before accessing any fields in this data structure the driver should always verify that the structure is valid by inspecting the signature field.

```

typedef struct _dev_info_t
{
    u32_t signature;
#define DEV_INFO_SIGNATURE          0x44564900
#define DEV_INFO_SIGNATURE_MASK     0xffffffff00
#define DEV_INFO_FEATURE_CFG_VALID 0x01
#define DEV_INFO_KEY_IN_EFFECT_MASK 0x06
#define DEV_INFO_MANUF_KEY_IN_EFFECT 0x02
#define DEV_INFO_UPGRADE_KEY_IN_EFFECT 0x04

```

```
#define DEV_INFO_NO_KEY_IN_EFFECT          0x06
#define DEV_INFO_DRV_ALWAYS_ALIVE           0x40
#define DEV_INFO_SECONDARY_PORT             0x80

shared_hw_cfg_t shared_hw_config;

/* 8 bits each: Major, minor, build, 0x05 */
u32_t bc_rev;

port_hw_cfg_t port_hw_config;
u32_t unused[4];

u32_t per_port_hw_config2;
u32_t l2_reserved2[3];

/* Format revision: applies to shared and port features */
u32_t format_rev;
#define FEATURE_FORMAT_REV_MASK            0xff000000
#define FEATURE_FORMAT_REV_ID              ('A' << 24)

shared_feat_cfg_t shared_feature_config;
port_feat_cfg_t port_feature_config;
port_feat_cfg_t port_feature_config2;

} dev_info_t;
```



SHARED HARDWARE CONFIGURATION (SHARED_HW_CFG)

The shared hardware configuration information applies to all ports of a NetXtreme II controller. Drivers may use the information in this area rather than reading the data from NVRAM connected to the NetXtreme II controller. For details on the meanings of these various settings, see [Section 3: “NVRAM Configuration” on page 35](#).

```
typedef struct _shared_hw_cfg_t
{
    u8_t part_num[16];
    u32_t power_dissipated;
        #define SHARED_HW_CFG_POWER_STATE_D3_MASK      0xff000000
        #define SHARED_HW_CFG_POWER_STATE_D2_MASK      0xff0000
        #define SHARED_HW_CFG_POWER_STATE_D1_MASK      0xff00
        #define SHARED_HW_CFG_POWER_STATE_D0_MASK      0xff
    u32_t power_consumed;
    u32_t config;
        #define SHARED_HW_CFG_DESIGN_NIC                0
        #define SHARED_HW_CFG_DESIGN_LOM                 0x1
        #define SHARED_HW_CFG_PHY_COPPER               0
        #define SHARED_HW_CFG_PHY_FIBER                0x2
        #define SHARED_HW_CFG_VAUX_OVERDRAW            0x4
        #define SHARED_HW_CFG_UMP_USE_MII              0
        #define SHARED_HW_CFG_UMP_USE_RMII             0x8
        #define SHARED_HW_CFG_WOL_ENABLE_BEACON         0x10
        #define SHARED_HW_CFG_PHY_FIBER_2_5G           0x20
        #define SHARED_HW_CFG_BACKPLANE_APP            0x40
        #define SHARED_HW_CFG_CRS_DV_SRC_SELECT_RXDV   0
        #define SHARED_HW_CFG_CRS_DV_SRC_SELECT_CRS    0x80
        #define SHARED_HW_CFG_LED_MODE_SHIFT_BITS     8
        #define SHARED_HW_CFG_LED_MODE_MASK            0x700
        #define SHARED_HW_CFG_LED_MODE_MAC             0
        #define SHARED_HW_CFG_LED_MODE_GPHY1          0x100
        #define SHARED_HW_CFG_LED_MODE_GPHY2          0x200
        #define SHARED_HW_CFG_LED_MODE_GPHY3          0x300
        #define SHARED_HW_CFG_LED_MODE_GPHY4          0x400
        #define SHARED_HW_CFG_LED_MODE_GPHY5          0x500
        #define SHARED_HW_CFG_LED_MODE_GPHY6          0x600
        #define SHARED_HW_CFG_LED_MODE_GPHY7          0x700
        #define SHARED_HW_CFG_UMP_PHY_TIMING_ENABLE   0x800
        #define SHARED_HW_CFG_UMP_PHY_TIMING_DISABLE  0
        #define SHARED_HW_CFG_MFW_CHOICE_SHIFT_BITS   12
        #define SHARED_HW_CFG_MFW_CHOICE_GPIO_MASK    0x7000
        #define SHARED_HW_CFG_MFW_CHOICE_IGNORE       0x0000
        #define SHARED_HW_CFG_MFW_CHOICE_UNUSED1     0x1000
        #define SHARED_HW_CFG_MFW_CHOICE_UNUSED2     0x2000
        #define SHARED_HW_CFG_MFW_CHOICE_GPIO3       0x3000
        #define SHARED_HW_CFG_MFW_CHOICE_GPIO4       0x4000
        #define SHARED_HW_CFG_MFW_CHOICE_GPIO5       0x5000
        #define SHARED_HW_CFG_MFW_CHOICE_GPIO6       0x6000
        #define SHARED_HW_CFG_MFW_CHOICE_GPIO7       0x7000
        #define SHARED_HW_CFG_GIG_LINK_ON_VAUX        0x8000
        #define SHARED_HW_CFG_LED_APP_MASK           0x30000
        #define SHARED_HW_CFG_LED_APP_INDEPENDENT    0x00000
        #define SHARED_HW_CFG_LED_APP_MULTI_COLOR    0x10000
        #define SHARED_HW_CFG_LED_APP_ALL_TIED      0x20000
    u32_t config2;
```

```
#define SHARED_HW_CFG2_NVM_SIZE_MASK          0xffff000
u32_t reserved[2];
} shared_hw_cfg_t;
```

PORT-SPECIFIC HARDWARE CONFIGURATION (PORT_HW_CFG)

The port-specific hardware configuration section is for hardware-specific features that apply to an individual port of the NetXtreme II. For details on the meanings for these various settings, see [Section 3: “NVRAM Configuration” on page 35](#).

```
#define PORT_HW_CFG_RESERVED_WORD_CNT 9
typedef struct _port_hw_cfg_t
{
    u32_t mac_upper;
        #define PORT_HW_CFG_UPPERMAC_MASK      0xffff
    u32_t mac_lower;
    u32_t config;
        #define PORT_HW_CFG_SERDES_TXCTL3_MASK 0xffff
        #define PORT_HW_CFG_DEFAULT_LINK_MASK 0x1f0000
        #define PORT_HW_CFG_DEFAULT_LINK_AN   0x0
        #define PORT_HW_CFG_DEFAULT_LINK_1G   0x30000
        #define PORT_HW_CFG_DEFAULT_LINK_2_5G 0x40000

    u32_t l2_reserved[7];
    u32_t iscsi_mac_upper; /* Upper 16 bits are always zeroes */
    u32_t iscsi_mac_lower;

    u32_t reserved[PORT_HW_CFG_RESERVED_WORD_CNT];
} port_hw_cfg_t;
```

SHARED FEATURE CONFIGURATION (SHARED_FEAT_CFG)

The shared feature configuration is used for configurable options that are shared across all ports on a single NetXtreme II controller. For details on the meanings for these various settings, refer to [Section 3: “NVRAM Configuration” on page 35](#).

```
typedef struct _shared_feat_cfg_t
{
    u32_t config;
        #define SHARED_FEATURE_RESERVED_MASK 0xffffffff
    u32_t reserved[3];
} shared_feat_cfg_t;
```

PORT FEATURE CONFIGURATION (PORT_FEAT_CFG)

The port feature configuration is used for configurable options that are applicable to an individual port of the NetXtreme II. For details on the meanings for these various settings, see [Section 3: “NVRAM Configuration” on page 35](#).

```

typedef struct _res_alloc_t
{
    u32_t version;
    #define RES_VER_STRING                      'A'
    #define RES_VER_STRING_MASK                 0xff000000
    #define RES_VER_STRING_SHIFT_BITS          24

    u32_t res_cfg;
    #define RES_RES_CFG_VALID                  0x01
    #define RES_RES_CFG_DIAG                 0x02
    #define RES_RES_CFG_L2                   0x04
    #define RES_RES_CFG_ISCSI                0x08
    #define RES_RES_CFG_RDMA                0x10

    u32_t enum_val;
    #define RES_ENUM_VALID                  RES_RES_CFG_VALID
    #define RES_ENUM_VAL_DIAG              RES_RES_CFG_DIAG
    #define RES_ENUM_VAL_L2                RES_RES_CFG_L2
    #define RES_ENUM_VAL_ISCSI              RES_RES_CFG_ISCSI
    #define RES_ENUM_VAL_RDMA              RES_RES_CFG_RDMA

    u32_t conn_resource1;
    #define RES_CONN_RDMA_MASK             0xfffff0000
    #define RES_CONN_TOE_MASK              0xfffff

    u32_t conn_resource2;
    #define RES_CONN_ISCSI_MASK            0xfffff0000
    #define RES_CONN_ISER_MASK              0xfffff

    u32_t conn_resource3;
    u32_t conn_resource4;

} res_alloc_t;

typedef struct _port_feat_cfg_t
{
    u32_t config;
    #define PORT_FEATURE_WOL_ENABLED        0x01000000
    #define PORT_FEATURE_MBA_ENABLED        0x02000000
    #define PORT_FEATURE_MFW_ENABLED        0x04000000
    #define PORT_FEATURE_RESERVED           0x08000000
    #define PORT_FEATURE_BAR1_SIZE_MASK     0xf
    #define PORT_FEATURE_BAR1_SIZE_DISABLED 0x0
    #define PORT_FEATURE_BAR1_SIZE_64K       0x1
    #define PORT_FEATURE_BAR1_SIZE_128K      0x2
    #define PORT_FEATURE_BAR1_SIZE_256K      0x3
    #define PORT_FEATURE_BAR1_SIZE_512K      0x4
    #define PORT_FEATURE_BAR1_SIZE_1M         0x5
    #define PORT_FEATURE_BAR1_SIZE_2M         0x6
    #define PORT_FEATURE_BAR1_SIZE_4M         0x7
    #define PORT_FEATURE_BAR1_SIZE_8M         0x8
    #define PORT_FEATURE_BAR1_SIZE_16M        0x9
    #define PORT_FEATURE_BAR1_SIZE_32M        0xa
    #define PORT_FEATURE_BAR1_SIZE_64M        0xb

```

```

#define PORT_FEATURE_BAR1_SIZE_128M           0xc
#define PORT_FEATURE_BAR1_SIZE_256M           0xd
#define PORT_FEATURE_BAR1_SIZE_512M           0xe
#define PORT_FEATURE_BAR1_SIZE_1G             0xf

u32_t wol_config;
#define FEATURE_WOL_DEFAULT_SHIFT_BITS        4
#define FEATURE_WOL_DEFAULT_MASK              0x30
#define FEATURE_WOL_DEFAULT_DISABLE          0
#define FEATURE_WOL_DEFAULT_MAGIC            0x10
#define FEATURE_WOL_DEFAULT_ACPI             0x20
#define FEATURE_WOL_DEFAULT_MAGIC_AND_ACPI   0x30
#define FEATURE_WOL_LINK_SPEED_MASK          0xf
#define FEATURE_WOL_LINK_SPEED_AUTONEG       0
#define FEATURE_WOL_LINK_SPEED_10HALF         1
#define FEATURE_WOL_LINK_SPEED_10FULL         2
#define FEATURE_WOL_LINK_SPEED_100HALF        3
#define FEATURE_WOL_LINK_SPEED_100FULL        4
#define FEATURE_WOL_LINK_SPEED_1000HALF       5
#define FEATURE_WOL_LINK_SPEED_1000FULL       6
#define FEATURE_WOL_LINK_SPEED_2500HALF       7
#define FEATURE_WOL_LINK_SPEED_2500FULL       8
#define FEATURE_WOL_AUTONEG_LIMIT_MASK        0xc0
#define FEATURE_WOL_AUTONEG_LIMIT_10           0x80
#define FEATURE_WOL_AUTONEG_LIMIT_100          0x00
#define FEATURE_WOL_AUTONEG_LIMIT_1000         0x40
#define FEATURE_WOL_AUTONEG_ADVERTISE_1000     0x40
#define FEATURE_WOL_RESERVED_PAUSE_CAP        0x400
#define FEATURE_WOL_RESERVED_ASYM_PAUSE_CAP   0x800

u32_t mba_config;
#define FEATURE_MBA_BOOT_AGENT_TYPE_SHIFT_BITS 0
#define FEATURE_MBA_BOOT_AGENT_TYPE_MASK        0x3
#define FEATURE_MBA_BOOT_AGENT_TYPE_PXE         0
#define FEATURE_MBA_BOOT_AGENT_TYPE_RPL         1
#define FEATURE_MBA_BOOT_AGENT_TYPE_BOOTP        2
#define FEATURE_MBA_BOOT_AGENT_TYPE_ISCSIB      3
#define FEATURE_MBA_LINK_SPEED_SHIFT_BITS       2
#define FEATURE_MBA_LINK_SPEED_MASK             0x3c
#define FEATURE_MBA_LINK_SPEED_AUTONEG          0
#define FEATURE_MBA_LINK_SPEED_10HALF           0x4
#define FEATURE_MBA_LINK_SPEED_10FULL           0x8
#define FEATURE_MBA_LINK_SPEED_100HALF          0xc
#define FEATURE_MBA_LINK_SPEED_100FULL          0x10
#define FEATURE_MBA_LINK_SPEED_1000FULL         0x18
#define FEATURE_MBA_SETUP_PROMPT_ENABLE        0x40
#define FEATURE_MBA_HOTKEY_CTRL_S              0
#define FEATURE_MBA_HOTKEY_CTRL_B              0x80
#define FEATURE_MBA_EXP_ROM_SIZE_SHIFT_BITS    8
#define FEATURE_MBA_EXP_ROM_SIZE_MASK          0xff00
#define FEATURE_MBA_EXP_ROM_SIZE_DISABLED      0
#define FEATURE_MBA_EXP_ROM_SIZE_1K             0x100
#define FEATURE_MBA_EXP_ROM_SIZE_2K             0x200
#define FEATURE_MBA_EXP_ROM_SIZE_4K             0x300
#define FEATURE_MBA_EXP_ROM_SIZE_8K             0x400
#define FEATURE_MBA_EXP_ROM_SIZE_16K            0x500
#define FEATURE_MBA_EXP_ROM_SIZE_32K            0x600
#define FEATURE_MBA_EXP_ROM_SIZE_64K            0x700

```

```

#define FEATURE_MBA_EXP_ROM_SIZE_128K          0x800
#define FEATURE_MBA_EXP_ROM_SIZE_256K          0x900
#define FEATURE_MBA_EXP_ROM_SIZE_512K          0xa00
#define FEATURE_MBA_EXP_ROM_SIZE_1M            0xb00
#define FEATURE_MBA_EXP_ROM_SIZE_2M            0xc00
#define FEATURE_MBA_EXP_ROM_SIZE_4M            0xd00
#define FEATURE_MBA_EXP_ROM_SIZE_8M            0xe00
#define FEATURE_MBA_EXP_ROM_SIZE_16M           0xf00
#define FEATURE_MBA_MSG_TIMEOUT_SHIFT_BITS     16
#define FEATURE_MBA_MSG_TIMEOUT_MASK          0xffff
#define FEATURE_MBA BIOS_BOOTSTRAP_SHIFT_BITS  20
#define FEATURE_MBA BIOS_BOOTSTRAP_MASK        0x300000
#define FEATURE_MBA BIOS_BOOTSTRAP_AUTO        0
#define FEATURE_MBA BIOS_BOOTSTRAP_BBS         0x100000
#define FEATURE_MBA BIOS_BOOTSTRAP_INT18H       0x200000
#define FEATURE_MBA BIOS_BOOTSTRAP_INT19H       0x300000

u32_t imd_config;
#define FEATURE_IMD_ECHO_MODE_ENABLE          0x2

u32_t mba_vlan_cfg;
#define FEATURE_MBA_VLAN_TAG_MASK            0xffff
#define FEATURE_MBA_VLAN_ENABLE              0x10000

res_alloc_t resource;

u32_t smbus_config;
#define FEATURE_SMBUS_ENABLE                1
#define FEATURE_SMBUS_ADDR_MASK             0xfe

u32_t iscsib_basic_config;
#define FEATURE_ISCSIB_SKIP_TARGET_BOOT     1

} port_feat_cfg_t;

```

SOFTWARE LICENSE KEY (LICENSE_KEY)

The NetXtreme II supports software licensing to enable advanced features such as RDMA, iSCSI, and an increased TOE connection offload limit. The software license area implements this functionality:

```

#define HASH_VALUE_SIZE                      12

typedef struct _license_key
{
    u8_t digest [HASH_VALUE_SIZE];
    /* KEY_VALID_PATTERN is used when no actual digest is needed */
    #define KEY_VALID_PATTERN_BYTE           0x5a
    #define KEY_VALID_PATTERN_DWORD         0x5a5a5a5a5a

    u8_t key_type;
    #define KEY_TYPE_ENUM_BCM5706           0x0
    u8_t version;
    #define VERSION_CURRENT                0x0
    u8_t dword_length;                     /* Not including the digest */
    u8_t oem_id;
    #define OEM_ID_BRCM                  0
    #define OEM_ID_HPQ                   0x3c

```

```

u16_t capability;
#define CAP_KERNEL_RDMA          0x0001
#define CAP_USER_RDMA            0x0002
#define CAP_TOE                  0x0004
#define CAP_ISCSI_INIT            0x0008
#define CAP_ISCSI_TRGT            0x0010
#define CAP_ISER_INIT              0x0020
#define CAP_ISER_TRGT              0x0040
#define CAP_ISCSI_BOOT              0x0080
#define CAP_ISCSI_FULL_ACCL        0x0100
#define CAP_ISCSI_HDR_DGST         0x0200
#define CAP_ISCSI_BODY_DGST         0x0400
u16_t max_toe_conn;
#define CONN_UNLIMITED           0xffff

u16_t reserved;
u16_t max_um_rdma_conn;

u16_t max_iscsi_init_conn;
u16_t max_iscsi_trgt_conn;

u16_t max_iser_init_conn;
u16_t max_iser_trgt_conn;

u32_t reserved_a[3];

u32_t sn;

u16_t reserved_b;
u16_t expiration;
#define EXPIRATION_NEVER          0xffff

} license_key_t;

#define FW_ENCODE_32BIT_PATTERN      0x1e1e1e1e
#define FW_ENCODE_16BIT_PATTERN       0x1e1e
#define FW_ENCODE_8BIT_PATTERN        0x1e

```

MANAGEMENT FIRMWARE STATE (MGMTFW_STATE)

The NetXtreme II supports running optional management firmware such as intelligent platform management initiative (IPMI) or UMP to provide additional functionality. The management firmware state is used to allow the host driver and the management firmware to interact. This field is not currently used but may be used in future software releases from Broadcom.

```

#define MGMTFW_STATE_WORD_SIZE 80
typedef struct _mgmtfw_state_t
{
    u32_t opaque [MGMTFW_STATE_WORD_SIZE];
} mgmtfw_state_t;

```

Section 9: Firmware Features

The NetXtreme II controller uses a combination of hardware, software, and firmware to implement all of its supported functionality. While some firmware is required for basic operation, other firmware is available as an option to provide enhanced functionality. This section focuses on the various features enabled through firmware and how these features can be used.

BOOTCODE

All implementations of the NetXtreme II controller require using bootcode. The bootcode is similar to the system BIOS in that it is loaded at power-on from an NVRAM device, it begins execution immediately, and it provides important runtime services to the host. The bootcode differs from the system BIOS in that it executes on the management processor (MCP) of the NetXtreme II controller and not on the host CPU. The bootcode is responsible for the following tasks at power-on:

- Programming the NetXtreme II primary MAC address
- Programming the NetXtreme II PCI vendor and device IDs
- Programming the NetXtreme II PCI subsystem vendor and device IDs
- Programming the NetXtreme II Base Address register (BAR) size
- Enabling any Wake-on-LAN (WOL) features as specified in NVRAM
- Performing any NetXtreme II errata fixes that may be required
- Loading optional management firmware (see UMP and IPMI later in this section)

The bootcode provides the following runtime services to the host:

- Handling expansion ROM accesses for implementations that include a PCI expansion ROM (such as Pre-boot eXecution Environment or PXE)
- Handling PCI vital product data (VPD) read accesses to the external NVRAM
- Performing license key validation
- Providing a driver/firmware handshake mechanism to coordinate hardware access between the OS device driver and the running firmware

The bootcode executes and initializes the NetXtreme II controller under the following conditions:

- When VAUX power is applied to the NetXtreme II controller
- When a PCI or power-on reset occurs
- When a local or GRC reset is performed by the OS device driver

For additional details on the configurable options that are supported by the bootcode, see [Section 3: "NVRAM Configuration" on page 35](#).

INTELLIGENT PLATFORM MANAGEMENT INITIATIVE

The NetXtreme II may be configured to include an optional firmware module to support the intelligent platform management initiative (IPMI). When appropriately configured in NVRAM, this firmware module is loaded by the bootcode and allows an external baseboard management controller (BMC) to access the Ethernet network over a system management bus (SMBus) interface. The protocol used by the BMC and the NetXtreme II is available from Broadcom as a separate application note and is not covered in this document. Contact your Broadcom representative for additional details on the optional IPMI firmware module and the NetXtreme II controllers that support the IPMI interface.

UNIVERSAL MANAGEMENT PORT

The NetXtreme II may be configured to include an optional firmware module to support the UMP. The UMP allows an external baseboard management controller (BMC) to access the Ethernet network over an MII or RMII type of interface. The protocol used by the BMC and the NetXtreme II is available from Broadcom as a separate application note and is not covered in this document. Contact your Broadcom representative for additional details on the optional UMP firmware module and the NetXtreme II controllers that support the UMP interface.

NETWORK CONTROLLER-SIDEband INTERFACE

The NetXtreme II may be configured to include an optional firmware module to support the DMTF's Network Controller-Sideband Interface (NC-SI). The NC-SI allows an external BMC to access the Ethernet network over an RMII interface. The protocol is very similar to UMP but is available as an open specification through the DMTF and is not covered in this document. Contact your Broadcom representative for additional details on the optional NC-SI firmware module and the NetXtreme II controllers that support the NC-SI interface.

Section 10: MAC Register Definitions

REGISTER ACCESS DEFINITIONS

[Table 50](#) describes the access modes for registers, along with the expected read and write behavior.

Table 50: Register Access Definitions

Mode	Description	Read Effect	Write Effect
RO	Read-Only	Returns current value	None
WO	Write-Only	Read as 0	Write 0 has no effect, write 1 starts command.
WC	Write-to-Clear	Returns current value	Write 0 has no effect, write 1 clears bit.
AC	Auto-Clear	Returns current value	None
SC	Self-Clearing	Returns 0 when action is complete; returns 1 while action is pending	Write 0 has no effect, write 1 starts action.
RW	Read-Write	Returns current value	Sets new value
RWS	Read-Write Sticky	Returns current value	Sets new value; value cannot be written again until after reset

REGISTER DEPENDENCY DEFINITIONS

Multiport controllers such as the BCM5709 include register definitions that are either shared between all ports of the controller (known as Common Core definitions) or are unique for each Ethernet port of the controller (known as TCP Offload Engine [or TOE] core definitions). The following definitions are used to describe the final bit or register setting when multiple Ethernet ports each have a register that accesses a common core.

Table 51: Register Dependency Definitions

Register	Description
SPLIT	When a bit definition includes a "[SPLIT]" designation it indicates that the resolved functionality of the bit is unique for each port. In other words, the bit is not shared between ports. Unless otherwise specified, all bit definitions are assumed to have a [SPLIT] designation.
OR	When a bit definition includes an "[OR]" designation it indicates that resolved functionality of the bit is the OR combination of the bit from all ports. Example: If port 0 sets the bit to 0 and port 1 sets the bit to 1, the resolved value will be 1. Only when all ports set the bit to 0 will the resolved value be 0. The "OR" designation only applies to single bits, it does not apply to multiple bit fields or entire registers.
AND	When a bit definition includes an "[AND]" designation it indicates that resolved functionality of the bit is the AND combination of the bit from all ports. Example: If port 0 sets the bit to 0 and port 1 sets the bit to 1, the resolved value will be 0. Only when all ports set the bit to 1 will the resolved value be 1. The "AND" designation applies to single bits as well as entire registers.

Table 51: Register Dependency Definitions (Cont.)

Register	Description
SHARE	When a bit or register definition includes a "[SHARE]" designation it indicates that the bit or register definition is shared by all ports of the device. Example: If port 0 sets the register to 0x01234567, then port 1 sets the register to 0x89abcdef, the resulting value will be the last value written or 0x89abcdef. If all ports write the register simultaneously, then the value from function 0 will be used.
SELECT	When a bit or register definition includes a "[SELECT]" designation it indicates that each bit or register is read or written independently. However, the PORT_SELECT field of the " Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68) " on page 172 will select which port will be used to resolve the bit or register function.
COMMON	When a bit or register definition includes a "[CMN]" designation it indicates that a Common Core reset will return the bit or register to its power-on default value.
TC	When a bit or register definition includes a "[TC]" designation it indicates that a TOE Core reset will return the bit or register to its power-on default value. Example: [TC0] indicates TOE Core 0, [TC1] indicates TOE core 1. All other TOE cores will not be affected by a TOE Core reset.

REGISTER RESET VALUES

While most registers have default values at power-on reset, some registers are undefined or depend on inputs sampled at power-on reset. For those registers that do not have fixed values at power-on reset, the default values are listed as 0XX throughout this documentation. In addition, the NetXtreme II controller may receive a reset from multiple sources. In most cases the register will return to its power-on reset value when any type of reset is asserted, but in some cases the register may retain its value unless a specific type of reset is asserted. The following table describes the three basic forms of reset. Unless otherwise specified, all registers will revert to their power-on reset value when any type of reset is asserted.

Table 52: Reset Forms

Reset Type	Reset Description
Hard	A Hard reset will occur when either the Vmain or Vaux power rails drop below their threshold values. This essentially means that a hard reset will occur when the NetXtreme II device loses power. (It should be noted that some devices such as the BCM5709 and BCM5716 support multiple "voltage islands" where Vmain and Vaux are not sourced from the same input and therefore only certain blocks will be affected by a Vmain or Vaux transition.)
PCI	A PCI reset will occur when any source of PCI or PCIe reset is asserted.
Core	A core reset will occur in response to a register write to reset the device such as when the CORE_RST_REQ bit of the " Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68) " on page 172 is set.

VARIABLE REGISTER DEFINITIONS

Different controllers in the NetXtreme II family may add registers or bit definitions, remove registers or bit definitions, or modify the meaning of an existing register or bit definition, depending on the controller and its revision. When there are differences between different controllers, the register definition table will include additional columns for each supported controller. Each column will include a designation to identify which controllers support that bit definition. An "X" in the controller column will indicate that all revisions of that controller support that bit, while an AX will indicate that all A steppings of that controller support that bit definition, and a B2 will indicate that only the B2 stepping of the controller supports that bit. When every controller beginning with C1 supports a particular bit definition then it will be marked as C1+. [Table 53](#) shows several examples of this layout.

Table 53: Example Register Bit Definitions

Bit	Name	Description	Mode	Reset	06	08	09
31	BIT_31	This field is only defined for the BCM5709 B-stepping controllers (including B0, B1, B2, etc.).	RW	0	-	BX	-
31	RESERVED	This field is defined for all controllers other than the BCM5709 BX (including A0, A1, A2..., C0, C1, C2..., D0, D1, D2,...).	RO	0	X	X	AX, C0+
30	BIT_30	This field is defined for all BCM5709 controllers.	RW	1	-	X	-
30	RESERVED	This field is defined for all BCM5706 and BCM5708 controllers.	RO	0	X	X	-
29-1	RESERVED	This field is defined the same for all NetXtreme II controllers.	RO	0	X	X	X
0	BIT_0	This field is defined for the BCM5706 A1 controller only.	RW	0	A1	-	-
0	RESERVED	This field is defined for all controllers other than the BCM5706 A1.	RO	0	A0, A2+	X	X

PCI CONFIGURATION BLOCK REGISTERS

VENDOR ID REGISTER (PCICFG_VENDOR_ID, OFFSET 0x00)

The read-only Vendor ID register identifies the manufacturer of the PCI adapter and defaults to 0x14e4 at power-on reset. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness.

This register may be modified by firmware (see “[PCI ID Value 1 Register \(pci_id_val1, Offset 0x434\)](#)” on page 211).

Table 54: Vendor ID Register (pcicfg_vendor_id, Offset 0x00)

Bit	Field	Description	Access	Reset	06	08	09	16
15–0	VENDOR_ID	Unique identifier for Broadcom.	RO	0x14e4	X	X	X	X

DEVICE ID REGISTER (PCICFG_DEVICE_ID, OFFSET 0x02)

The read-only Device ID register identifies a particular adapter within those made by the same manufacturer.

This register may be modified by firmware (see “[PCI ID Value 1 Register \(pci_id_val1, Offset 0x434\)](#)” on page 211).

Table 55: Device ID Register (pcicfg_device_id, Offset 0x02)

Bit	Field	Description	Access	Reset
15–0	DEVICE_ID	Unique identifier for the BCM5706C MAC Transceiver.	RO	0x164a
		Unique identifier for the BCM5706S MAC Transceiver.		0x16aa
		Unique identifier for the BCM5708C MAC Transceiver		0x164c
		Unique identifier for the BCM5708S MAC Transceiver		0x16ac
		Unique identifier for the BCM5709C MAC Transceiver		0x1639
		Unique identifier for the BCM5709S MAC Transceiver		0x163a
		Unique identifier for the BCM5716 MAC Transceiver		0x163b

COMMAND REGISTER (PCICFG_COMMAND, OFFSET 0x04)

The read-write Command register is used to enable various features of the device. All of the bit positions are predefined by the PCI specification. Not all bits defined in the PCI specification are implemented by this register.

Table 56: Command Register (pcicfg_command, Offset 0x04)

Bit	Field	Description	Mode	Reset	06	08	09	16
15–11	RESERVED	RESERVED	RO	0x0	X	X	X	X
10	INT_DISABLE	When this bit is set, the interrupt output will be masked (deasserted) regardless of any internal chip logic. Setting this bit has no effect on the INT_STATUS bit below. Clearing this bit will unmask the interrupt and let it run normally.	RW	0	X	X	X	X
9	FAST_B2B	This bit is hardwired to 0 to indicate that this device does not generate fast back-to-back cycles to different devices on the PCI bus.	RO	0	X	X	–	–
9	RESERVED		RO	0	–	–	X	X
8	SERR_ENA	When this bit is set, the device will drive the SERR signal when a parity error is detected during an address cycle.	RW	0	X	X	–	–
8	SERR_ENA	When this bit is set, fatal and non-fatal errors detected by the controller are reported to the Root Complex.	RW	0	–	–	X	X
7	STEPPING	This bit is hardwired to 0 to indicate that stepping control is not supported.	RO	0	X	X	X	X
6	PERR_ENA	When this bit is set, the device will drive the PERR signal when a parity error is detected.	RW	0	X	X	–	–
6	PERR_ENA	When this bit is set the MSTR_PERR bit of the “Status Register (pcicfg_status, Offset 0x06)” on page 154 is set when the requester receives a poisoned Completion of the requester poisons a write request.	RW	0	–	–	X	X
5–3	RESERVED		RO	0	–	–	X	X
5	VGA_SNOOP	This bit is hardwired to 0 to indicate that this device does not support VGA palette snoop.	RO	0	X	X	–	–
4	MWI_CYCLES	When this bit is set, Memory Write Invalidate cycles may be generated under certain circumstances.	RW	0	X	X	–	–
3	SPECIAL_CYCLES	This bit is hardwired to 0 to indicate that this device does not respond to special cycles.	RW	0	X	X	–	–
2	BUS_MASTER	This bit controls the enabling of the PCI bus master activity by this device.	RW	0	X	X	X	X
1	MEM_SPACE	This bit controls the enabling of the PCI memory space. When enabled, PCI accesses that map to the BAR register values will be mapped to the PCI Normal Mode Address Space.	RW	0	X	X	X	X
0	IO_SPACE	This bit indicates that the device does not support I/O space access because it is 0 and cannot be modified.	RO	0	X	X	X	X

STATUS REGISTER (PCICFG_STATUS, OFFSET 0x06)*Table 57: Status Register (pcicfg_status, Offset 0x06)*

Bit	Name	Description	Mode	Reset	06	08	09	16
15	PAR_ERR	When this bit is set, it indicates that the device has detected a parity error regardless of the setting of the PERR_ENA bit. This bit is cleared by writing a 1 to this position. Writing a 0 to this position has no effect.	WC	0	X	X	-	-
15	PAR_ERR	When this bit is set it indicates that the function has received a poisoned TLP.	WC	0	-	-	X	X
14	SIG_SERR	When this bit is set, it indicates that the device has asserted a SERR due to a parity error on an address phase. This bit is cleared by writing a 1 to this position. Writing a 0 to this position has no effect.	WC	0	X	X	X	X
13	RCV_MSTR_ABT	When this bit is set, it indicates that the device received a master abort while it was the master on the bus. This means that no slave responded to the address generated on the bus. This bit is cleared by writing a 1 to this position. Writing a 0 to this position has no effect.	WC	0	X	X	X	X
12	RCV_TGT_ABT	When this bit is set, it indicates that the device indicate a target abort due to the normal reason when the master asserted STOP. This bit is cleared by writing a 1 to this position. Writing a 0 to this position has no effect.	WC	0	X	X	X	X
11	SIG_TGT_ABT	When this bit is set, it indicates that the device indicate a target abort due to the master going away rather than due to the data not being available. This indicates that the master disappeared in the middle of the cycle. This bit is cleared by writing a 1 to this position. Writing a 0 to this position has no effect.	WC	0	X	X	X	X
10–9	RESERVED		RO	0	-	-	X	X
10–9	DEVSEL_TIMING	These bits encode the slowest timing of DEVSEL, except for configuration cycles. Value entries are 00 for fast, 01 for medium and 10 for slow. The device is hardwired for medium DEVSEL response speed.	RO	0x1	X	X	-	-
8	SIG_PERR	When this bit is set, it indicates that a parity error was detected while the device was a master on the bus and while parity error reporting was enabled (see PERR_ENA). This bit is cleared by writing a 1 to this position. Writing a 0 to this position has no effect.	WC	0	X	X	X	X
7	RESERVED		RO	0	-	-	X	X
7	FAST_B2B_CAP	This bit is tied high to indicate that the device supports reception of fast back-to-back cycles to different devices on the bus.	RO	0x1	X	X	-	-
6	RESERVED		RO	0	X	X	X	X
5	RESERVED		RO	0	-	-	X	X

Table 57: Status Register (pcicfg_status, Offset 0x06) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
5	66MHZ_CAP	This bit is tied high to indicate that the device supports 66-MHz operation.	RO	0x1	X	X	—	—
4	CAP_LIST	This bit is tied high to indicate that the device supports a capability list. The list starts at address 0x40.	RO	0x1	X	X	X	X
3	INT_STATUS	This bit indicates the internal interrupt request state (before being masked by INT_DISABLE). A 0 indicates that no interrupt is pending. A 1 indicates that there is an interrupt pending.	RO	0	X	X	X	X
2–0	RESERVED		RO	0	X	X	X	X

REVISION ID REGISTER (PCICFG_REVISION_ID, OFFSET 0x08)

Description	Mode	Reset	06	08	09	16
The Revision ID register identifies the revision of the PCI adapter. This register may be modified by firmware (see “PCI ID Value 3 Register (pci_id_val3, Offset 0x43c)” on page 213).	RO	0	X	X	X	X

CLASS CODE REGISTER (PCICFG_CLASS_CODE, OFFSET 0x09)

Description	Mode	Reset	06	08	09	16
The Class Code register identifies the generic function of the device. This register may be modified by firmware (see “PCI ID Value 3 Register (pci_id_val3, Offset 0x43c)” on page 213).	RO	0x200000	X	X	X	X

CACHE LINE SIZE REGISTER (PCICFG_CACHE_LINE_SIZE, OFFSET 0x0c)

Description	Mode	Reset	06	08	09	16
The Cache Line Size register is used by the PCI-based host to indicate the size of a cache line in 32-bit increments. Cache Line Sizes of 8, 16, 32, 64, 128, 256, and 512 bytes are supported. This field is used to determine the read command. Memory Read Multiple is sent if cache_line_size size is not 0 and memory transfer will cross cacheline boundary. Memory Read Line is sent if cache_line_size size is not 0 and memory transfer is greater than 1DW and transfer will not cross cacheline boundary. Memory Read is sent if 64 bits or less is transferred or cacheline size is 0.	RO	0	X	X	—	—
The Cache Line Size register is not used by PCI Express devices.	RO	0	—	—	X	X

LATENCY TIMER REGISTER (PCICFG_LATENCY_TIMER, OFFSET 0x0D)

Description	Mode	Reset	06	08	09	16
The read-write Latency Timer register is used by the PCI-based host to indicate the number of PCI clocks in which the chip may own the bus before checking to see if the bus should be relinquished. Only the upper 5-bits are usable as the lower 3-bits are hardwired to 0. This register is set to 0x00 or 0x40 at reset for conventional PCI or PCI-X, respectively.	RW	0xXX	X	X	-	-
The Latency Timer register is not used by PCI Express devices.	RW	0	-	-	X	X

HEADER TYPE REGISTER (PCICFG_HEADER_TYPE, OFFSET 0x0E)

Description	Mode	Reset	06	08	09	16
The Header Type register identifies both the layout of bytes 0x10 through 0x3f of the Configuration space, as well as whether this adapter contains multiple functions. This register is always 0x00, which indicates a single function device (Type 0) using the format specified in the PCI specification.	RO	0x00	X	X	-	-
The Header Type register identifies both the layout of bytes 0x10 through 0x3f of the configuration space, as well as whether this adapter contains multiple functions. This register is always 0x80 which indicates a multi-function device (Type 0) using the format specified in the PCI specification.	RO	0X80	-	-	X	X

BIST REGISTER (PCICFG_BIST, OFFSET 0x0F)

Description	Mode	Reset	06	08	09	16
The BIST register is used to initiate and report the results of any Built-In-Self-Test. This register may be modified by firmware (see " PCI ID Value 6 Register (pci_id_val6, Offset 0x44c) " on page 219).	RO	0	X	X	X	X

BASE ADDRESS 1 REGISTER (PCICFG_BAR_1, OFFSET 0x10)

The 32-bit Base Address 1 register programs the base address for the memory space mapped by the card onto the PCI bus. This register can be combined with the Base Address 2 register to make a 64-bit address for supporting Dual Address cycle systems.

Table 58: Base Address 1 Register (pcicfg_bar_1, Offset 0x10)

Bit	Name	Description	Mode	Reset
31–4	ADDRESS	These bits set the address within a 32-bit address space that will be card will respond in. These bits may be combined with the bits in BAR_2 to create a full 64-bit address decode. Only the bits that address blocks bigger than the setting in the BAR1_SIZE value are RW. All lower bits are RO with a value of 0. This value is sticky and is only reset by a HARD reset. This value can be modified by firmware (see " PCI ID Value 6 Register (pci_id_val6, Offset 0x44c) " on page 219).	RW	0
3	PREFETCH	This bit indicates that the area mapped by BAR_1 may not be pre-fetched or cached by the system without side effects.	RO	0



Table 58: Base Address 1 Register (pcicfg_bar_1, Offset 0x10) (Cont.)

Bit	Name	Description	Mode	Reset
2-1	SPACE_TYPE	These bits indicate that BAR_1 may be programmed to map this adapter to anywhere in the 64-bit address space.	RO	0x2
0	MEM_SPACE	This bit indicates that BAR_1 maps a memory space and is always read as 0.	RO	0

BASE ADDRESS 2 REGISTER (PCICFG_BAR_2, OFFSET 0x14)

The 32-bit Base Address 2 register programs the upper half of the base address for the memory space mapped by the card onto the PCI bus.

Table 59: Base Address 2 Register (pcicfg_bar_2, Offset 0x14)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	ADDR	These bits set the address upper 32-bit address space that will be card will respond in. These bits may be combined with the bits in Base Address 1 register to create a full 64-bit address decode. These bits must be set to 0 for the card to respond to single address cycle requests. This value is sticky and is only reset by a hard reset.	RW	0	X	X	X	X

BASE ADDRESS 3REGISTER (PCICFG_BAR_3, OFFSET 0x18)

The 32-bit Base Address 3 register programs the base address for the memory space mapped by the card onto the PCI bus. This register can be combined with the Base Address 2 register to make a 64-bit address for supporting Dual Address cycle systems.

Table 60: Base Address 3 Register (pcicfg_bar_3, Offset 0x18)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-4	ADDRESS	These bits set the address within a 32-bit address space that will be card will respond in. These bits may be combined with the bits in BAR_4 to create a full 64-bit address decode. Only the bits that address blocks bigger than the setting in the BAR3_SIZE value are RW. All lower bits are RO with a value of 0. This value is sticky and is only reset by a HARD reset.	RO	0:Hard	X	X	-	-

Table 60: Base Address 3 Register (pcicfg_bar_3, Offset 0x18) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–4	ADDRESS	These bits set the address within a 32-bit address space that will be card will respond in. These bits may be combined with the bits in BAR_4 to create a full 64-bit address decode. Only the bits that address blocks bigger than the setting in the BAR3_SIZE value are RW. All lower bits are RO with a value of 0. This value is sticky and is only reset by a HARD reset. This value can be modified by firmware (see “PCI Base Address 2 Configuration Register (pci_bar2_config, offset 0x4e0)” on page 371).	RW	0:Hard	—	—	X	X
3	PREFETCH	This bit indicates that the area mapped by BAR_3 may not be pre-fetched or cached by the system without side effects.	RO	0	—	—	X	X
2–1	SPACE_TYPE	These bits indicate that BAR_3 may be programmed to map this adapter to anywhere in the 64-bit address space.	RO	0x0	—	—	X	X
0	MEM_SPACE	This bit indicates that BAR_3 maps a memory space and is always read as 0.	RO	0	—	—	X	X

BASE ADDRESS 4 REGISTER (PCICFG_BAR_4, OFFSET 0x1C)

The 32-bit Base Address 4 register programs the upper half of the base address for the memory space mapped by the card onto the PCI bus.

Table 61: Base Address 4 Register (pcicfg_bar_4, Offset 0x14)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	ADDR	These bits set the address upper 32-bit address space that will be card will respond in. These bits may be combined with the bits in Base Address 3 register to create a full 64-bit address decode. These bits must be set to 0 for the card to respond to single address cycle requests. This value is sticky and is only reset by a hard reset.	RO	0:Hard	X	X	—	—
31–0	ADDR	These bits set the address upper 32-bit address space that will be card will respond in. These bits may be combined with the bits in Base Address 3 register to create a full 64-bit address decode. These bits must be set to 0 for the card to respond to single address cycle requests. This value is sticky and is only reset by a hard reset.	RW	0:Hard	—	—	X	X

SUBSYSTEM VENDOR ID REGISTER (PCICFG_SUBSYSTEM_VENDOR_ID, OFFSET 0x2C)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
The read-only Subsystem Vendor ID register is used by the adapter manufacturer for identification and has a value of 0x14e4 at power-on reset. This value can be modified by firmware (see “PCI ID Value 2 Register (pci_id_val2, Offset 0x438)” on page 212).	RO	0x14e4	X	X	X	X

SUBSYSTEM ID REGISTER (PCICFG_SUBSYSTEM_ID, OFFSET 0x2E)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
The Subsystem ID register is used by the adapter manufacturer for identification. This value can be modified by firmware (see “PCI ID Value 2 Register (pci_id_val2, Offset 0x438)” on page 212).	RO	0x164a	X	—	—	—
The Subsystem ID register is used by the adapter manufacturer for identification. This value can be modified by firmware (see “PCI ID Value 2 Register (pci_id_val2, Offset 0x438)” on page 212).	RO	0x164c	—	X	—	—
The Subsystem ID register is used by the adapter manufacturer for identification. This value can be modified by firmware (see “PCI ID Value 2 Register (pci_id_val2, Offset 0x438)” on page 212).	RO	0x1639	—	—	X	X

EXPANSION ROM BASE ADDRESS REGISTER (PCICFG_EXP_ROM_BAR, OFFSET 0x30)

The 32-bit Expansion ROM Base Address register programs the base address for the memory space mapped by the chip for use as an expansion ROM.

Table 62: Expansion ROM Base Address Register (pcicfg_exp_rom_bar, Offset 0x30)

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	18
31–1	ADDRESS	These bits indicate the address of the Expansion ROM area. This value can be modified by firmware (see “PCI ID Value 2 Register (pci_id_val2, Offset 0x438)” on page 212).	RW/ RO	0xXX	X	X	X	X
0	BAR_ENA	This bit indicates that the Expansion ROM BAR is valid when set to 1. If it is 0, the expansion BAR should not be programmed or used. This value can be modified by firmware (see EXP_ROM_SIZE, “PCI ID Value 2 Register (pci_id_val2, Offset 0x438)” on page 212).	R/W	0	X	X	X	X

CAPABILITIES POINTER REGISTER (PCICFG_CAP_POINTER, OFFSET 0x34)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The Capabilities Pointer register specifies an Offset in the PCI address space of a linked list of new capabilities. The capabilities PCI-X, PCI Power Management, Vital Product Data (VPD), and Message Signaled Interrupts (MSI) are supported. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440)" on page 214).	RO	0x40	X	X	-	-
The Capabilities Pointer register specifies an Offset in the PCI address space of a linked list of new capabilities. The capabilities PCI Power Management, Vital Product Data (VPD), Message Signaled Interrupts (MSI), MSI-X, and PCI Express are supported. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440)" on page 214).	RO	0x48	-	-	X	X

INTERRUPT LINE REGISTER (PCICFG_INT_LINE, OFFSET 0x3C)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The Interrupt Line register is used to communicate interrupt line routing information. This field is set by the host and later used by any driver which is required to know which physical interrupt on the system interrupt controller is assigned to this device.	RW	0xXX	X	X	X	X

INTERRUPT PIN REGISTER (PCICFG_INT_PIN, OFFSET 0x3D)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The Interrupt Pin register is used to indicate which interrupt pin the device uses.	RO	0x1	X	X	X	X

MINIMUM GRANT REGISTER (PCICFG_MIN_GRANT, OFFSET 0x3E)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The Minimum Grant register is used to indicate the device's desired minimum grant period in units of 250 ns, assuming a PCI clock rate of 33 MHz. Devices should specify values that will allow them to most effectively use their internal resources as well as the PCI bus. This value can be modified by firmware (see " PCI ID Value 6 Register (pci_id_val6, Offset 0x44c) " on page 219).	RO	0	X	X	-	-
The Minimum Grant register is not supported for PCI Express. This value can be modified by firmware (see " PCI ID Value 6 Register (pci_id_val6, Offset 0x44c) " on page 219).	RO	0	-	-	X	X

MAXIMUM LATENCY REGISTER (PCICFG_MAXIMUM_LATENCY, OFFSET 0x3F)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The Maximum Latency register is used to indicate the device's desired maximum time between being granted the PCI bus in units of 250 ns, assuming a PCI clock rate of 33 MHz. Devices should specify values that will allow them to most effectively use their internal resources. This value can be modified by firmware (see " PCI ID Value 6 Register (pci_id_val6, Offset 0x44c) " on page 219).	RO	0	X	X	-	-
The Maximum Latency register is not supported for PCI Express. This value can be modified by firmware (see " PCI ID Value 6 Register (pci_id_val6, Offset 0x44c) " on page 219).	RO	0	-	-	X	X

PCI-X CAPABILITIES REGISTERS**PCI-X CAPABILITIES REGISTER (PCICFG_PCIX_CAP_ID, OFFSET 0x40)**

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The PCI-X Capability ID register indicates that the next 8 bytes contain a PCI-X RO capabilities block. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440) " on page 214).	RO	0x7	X	X	-	-

PCI-X NEXT CAPABILITY POINTER REGISTER (PCICFG_PCIX_NEXT_CAP_PTR, OFFSET 0x41)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The Next Capability register continues the PCI capability chain. Its value specifies an offset in the PCI address space of the next capability. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440) " on page 214).	RO	0x48	X	X	-	-

PCI-X COMMAND REGISTER (PCICFG_PCIX_COMMAND, OFFSET 0x42)*Table 63: PCI-X Command Register (pcicfg_pcix_command, Offset 0x42)*

Bit	Name	Description	Mode	Reset	06	08	09	16
15–7	RESERVED		RO	0	X	X	X	X
6–4	MAX_SPLIT	Sets the maximum number of Split Transactions the device is permitted to have outstanding at one time.	RO	0	X	X	–	–
			Value	Name	Description			
			0	MAX_SPLIT_1	1 Split Transaction			
			1	MAX_SPLIT_2	2 Split Transactions			
			2	MAX_SPLIT_3	3 Split Transactions			
			3	MAX_SPLIT_4	4 Split Transactions			
			4	MAX_SPLIT_8	8 Split Transactions			
			5	MAX_SPLIT_12	12 Split Transactions			
			6	MAX_SPLIT_16	16 Split Transactions			
			7	MAX_SPLIT_32	32 Split Transactions			
			0x1ff	RESERVED	RESERVED			
6–0	RESERVED		RO	0	–	–	X	X
3–2	MAX_MEM_READ	Sets the maximum byte count the device uses when initiating a Sequence with one of the burst memory read commands. The chip is capable of bursting up to 4096 bytes.	RW	0	X	X	–	–
			Value	Name	Description			
			0	512	Limit to 512 Bytes Max Read Size			
			1	1K	Limit to 1024 Bytes Max Read Size			
			2	2K	Limit to 2048 Bytes Max Read Size			
			3	4K	Limit to 4096 Bytes Max Read Size			
1	RELAX_ORDER	When set, the device is permitted to set the Relaxed Ordering bit in the Requestor Attributes of transactions it initiates that do not require strong write ordering.	RW	0x1	X	X	–	–
0	DATA_PAR_ERR	When set, the device should attempt to recover from data parity errors. If this bit is not set and the device is in PCI-X mode, the device asserts SERR (if enabled) whenever the Master Data Parity Error bit (bit 8 of Status Register) is set.	RW	0	X	X	–	–

PCI-X STATUS REGISTER (PCICFG_PCIX_STATUS, OFFSET 0x44)*Table 64: PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–30	RESERVED		RO	0	X	X	X	X
29	SPLIT_ERR	Indicates the device received a split completion message with the Split Completion Error attribute set. This bit is cleared by writing a 1 to this bit position.	WC	0	X	X	—	—
29–0	RESERVED		RO	0	—	—	X	X
28–26	MAX_CUM_SIZE	Indicates a number that is greater than or equal to the maximum cumulative size of all burst memory read transactions the device is designed to have outstanding at one time. This value is controlled from the PCI register space by the (MAX_CUMULATIVE_SIZE) value.	RO	0	X	X	—	—
Value	Name	Description						
0	1 KB	1-KB Max Outstanding Read Data						
1	2 KB	2-KB Max Outstanding Read Data						
2	4 KB	4-KB Max Outstanding Read Data						
3	8 KB	8-KB Max Outstanding Read Data						
4	16 KB	16-KB Max Outstanding Read Data						
5	32 KB	32-KB Max Outstanding Read Data						
6	64 KB	64-KB Max Outstanding Read Data						
7	128 KB	128-KB Max Outstanding Read Data						

Table 64: PCI-X Status Register (*pcicfg_pcix_status*, Offset 0x44) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25–23	MAX_SPLIT	Indicates a number greater than or equal to the maximum number of Split Transactions the device is designed to have outstanding at one time. This bit is controlled by the MAX_SPLIT_SIZE value in the PCI register space.	RO	0	X	X	–	–
		Value	Name	Description				
		0	1	Chip Supports 1 Split Transaction				
		1	2	Chip Supports 2 Split Transactions				
		2	3	Chip Supports 3 Split Transactions				
		3	4	Chip Supports 4 Split Transactions				
		4	8	Chip Supports 8 Split Transactions				
		5	12	Chip Supports 12 Split Transactions				
		6	16	Chip Supports 16 Split Transactions				
		7	32	Chip Supports 32 Split Transactions				
22–21	MAX_MEM_READ	Indicates a number greater than or equal to the maximum byte count the device is designed to use when initiating a Sequence with one of the burst memory read commands. This bit is controlled by the MAX_MEM_READ_SIZE value in the PCI register space.	RO	0	X	X	–	–
		Value	Name	Description				
		0	512	Chips Supports 512-Bytes Max Read Size				
		1	1K	Chips Supports 1024-Bytes Max Read Size				
		2	2K	Chips Supports 2048-Bytes Max Read Size				
		3	4K	Chips Supports 4096-Bytes Max Read Size				
20	DEV_COMPLEX	This bit indicates whether this device is a simple device or a bridge device. This chip is a simple device, therefore this bit is hardwired to 0.	RO	0	X	X	–	–
19	UNEXPECTED_SPLIT	This bit is set if an unexpected Split Completion with this device's Requester ID is received. This bit is cleared by writing a 1 to this bit position.	WC	0	X	X	–	–
18	SPLIT_DISCARD	This bit is set if the device discards a Split Completion because the requester would not accept it. This bit is cleared by writing a 1 to this bit position.	WC	0	X	X	–	–

Table 64: PCI-X Status Register (pcicfg_pcix_status, Offset 0x44) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
17	133_MHZ	This bit indicates that the device is capable of 133-MHz operation in PCI-X Mode. This bit is controlled by the 64_BIT_ADVERTISE bit in the PCI register space.	RO	0x1	X	X	-	-
16	64_BIT	This bit indicates the size of the device's AD bus. This bit is controlled by the 64_BIT_ADVERTISE bit in the PCI register space.	RO	0x1	X	X	-	-
15–8	BUS_NUM	This field indicates the number of the bus containing this function. This is mapped to bits AD[7–0] of the last configuration transaction received by this function. This is used for diagnostics only.	RO	0xff	X	X	-	-
7–3	DEV_NUM	This field indicates the number of the device containing this function. This is mapped to bits AD[15–11] of the last configuration transaction received by this function. This is used for diagnostics only.	RO	0x1f	X	X	-	-
2–0	FUNC_NUM	Function Number This field indicates the value of the function number of this function. This is used for diagnostics only.	RO	0x1	X	X	-	-

PCI POWER MANAGEMENT REGISTERS

POWER MANAGEMENT CAPABILITY ID REGISTER (PCICFG_PM_CAP_ID, OFFSET 0x48)

Description	Mode	Reset	06	08	09	16
The Power Management Capability ID register indicates that the next 8 bytes contain a Power Management capabilities block. This value can be modified by firmware (see “PCI ID Value 4 Register (pci_id_val4, Offset 0x440)” on page 214).	RO	0x1	X	X	X	X

POWER MANAGEMENT NEXT CAPABILITY REGISTER (PCICFG_PM_NEXT_CAP_PTR, 0x49)

Description	Mode	Reset	06	08	09	16
The Next Capability register continues the PCI capability chain. Its value specifies an RO offset in the PCI address space of the next capability. This value can be modified by firmware (see “PCI ID Value 4 Register (pci_id_val4, Offset 0x440)” on page 214).	RO	0x50	X	X	X	X

POWER MANAGEMENT CAPABILITIES REGISTER (PCICFG_PM_CAPABILITY, OFFSET 0x4A)**Table 65: Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)**

Bit	Name	Description	Mode	Reset	06	08	09	16
15	PME_IN_D3_COLD	This bit indicates whether the device supports asserting PME from the D3 _{cold} power state. This is supported if the VAUX_PRESENT input pin is high. This bit reflects the input value of the VAUX_PRESENT input pin.	RO	X	X	X	X	X
14	PME_IN_D3_HOT	This bit indicates whether the device supports asserting PME from the D3 _{hot} power state. This bit is controlled by the PME_IN_D3_HOT bit in the PCI register space.	RO	0x1	X	X	X	X
13	PME_IN_D2	This bit indicates whether the device supports asserting PME from the D2 power state. This bit is controlled by the PME_IN_D2 bit in the PCI register space.	RO	0	X	X	X	X
12	PME_IN_D1	This bit indicates whether the device supports asserting PME from the D1 power state. This bit is controlled by the PME_IN_D1 bit in the PCI register space.	RO	0	X	X	X	X
11	PME_IN_D0	This bit indicates whether the device supports asserting PME from the D0 power state. This bit is controlled by the PME_IN_D0 bit in the PCI register space.	RO	0	X	X	X	X
10	D2_SUPPORT	This bit indicates whether the device supports the D2 power management state. This bit is controlled by the D2_SUPPORT bit in the PCI register space.	RO	0	X	X	X	X
9	D1_SUPPORT	This bit indicates whether the device supports the D1 power management state. This bit is controlled by the D1_SUPPORT bit in the PCI register space.	RO	0	X	X	X	X
8–6	AUX_CURRENT	These bits report the 3.3Vaux auxiliary current requirements for the device. This chip uses the Data Register feature for this so this field is hardwired to 0.	RO	0	X	X	X	X
5	DSI	This bit indicates that the device requires a specific initialization (DSI) sequence following a transition to the D0 uninitialized state. This device does not need this support, so the bit is hardwired to 0.	RO	0	X	X	X	X
4	RESERVED		RO	0	X	X	X	X
3	CLOCK	This bit indicates that the device relies on the presence of the PCI clock for PME operation. This chip does not require the PCI clock to generate PME, therefore this bit is hardwired to 0.	RO	0	X	X	X	X
2	RESERVED		RO	0	X	X	X	X
1–0	VERSION	These bits indicate that this device complies with revision 1.1 of the PCI Power Management Interface Specification.	RO	0x2	X	X	–	–
1–0	VERSION	These bits indicate that this device complies with revision 1.2 of the PCI Power Management Interface specification.	RO	0x3	–	–	X	X

POWER MANAGEMENT CONTROL/STATUS REGISTER (PCICFG_PM_CSR, OFFSET 0x4C)*Table 66: Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
15	PME_STATUS	This bit is set when a PME is asserted from the MAC or RX Parser blocks, regardless of the state of the PME_ENABLE bit. If both this bit and the PME_ENABLE bit are high, then the PME output will be asserted low. This bit is cleared by writing a 1 in this bit position. At power-up, the chip must clear this bit, but on assertions of PCI_RST after that, this bit is sticky and not modified.	WC	0	X	X	X	X
14–13	DATA_SCALE	These bits indicate the scaling factor to be used when interpreting the values in the PM data register. The hardware default value for this field is 0x1, but this value can be written by firmware through the PCI register space (SCALE_PRG) to modify the read value to the host.	RO	0x1	X	X	X	X
12–9	DATA_SEL	These bits select which data is to be reported through the pm_data register. (Offset 0x4f) Select values other than those listed cause the pm_data register to return 0.	RW	0	X	X	X	X
8	PME_ENABLE	This bit enables the device to assert PME. At power-up, the chip must clear this bit, but on assertions of PCI_RST after that, this bit is sticky and not modified. On HARD reset, this bit resets to 1. On CORE reset, this bit resets to the value of the VAUX_PRSNT pin.	RW	X	X	X	X	X

Table 66: Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7–4	RESERVED	RESERVED	RO	0	X	X	X	X
3	RESERVED	RESERVED	RO	0x0	X	X	–	–
3	NO_SOFT_RESET	When this bit is set it indicates that when the device transitions from D3 to D0 it does not perform an internal reset.	RO	1	–	–	X	X
2	RESERVED	RESERVED	RO	0x0	X	X	X	X
1–0	STATE	These bits may be used by the system to set the power state. The register is implemented as two banks of two bits each. Can be written from both configuration space and from the PCI register space as the PM_STATE bits. When written from the PCI bus, only values of 0 and 3 will be accepted. This is the register returned on reads of this register from configuration space. The second bank catches all writes values. The value of the second register is returned when the PM_STATE bits are read from register space.	RW	0	X	X	X	X
Value	Name	Description						
0	D0	Select D0						
1	D1	Select D1. Stored only in the PM_STATE read value register.						
2	D2	Select D2. Stored only in the PM_STATE read register.						
3	D3_HOT	Select D3						

POWER MANAGEMENT C/S BSE REGISTER (PCICFG_PM_CSR_BSE, OFFSET 0x4E)

Description	Mode	Reset	06	08	09	16
The Power Management C/S BSE Register (PMCSR PCI to PCI Bridge Support Extensions) is not supported.	RO	0	X	X	X	X

POWER MANAGEMENT DATA REGISTER (PCICFG_PM_DATA, OFFSET 0x4F)

Description	Mode	Reset	06	08	09	16
The Power Management Data register returns one of eight different values selected RO by the DATA_SEL bits of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167. The reset value of all 8 different selections is 0. These values can be modified by firmware (see “PCI Power Management Data A Register (pci_pm_data_a, Offset 0x410)” on page 208 and “PCI Power Management Data B Register (pci_pm_data_b, Offset 0x414)” on page 209 and “PCI Power Management Data C Register (pci_pm_data_c, offset 0x46c)” on page 366).	0		X	X	X	X

VITAL PRODUCT DATA CAPABILITIES REGISTERS

VPD CAPABILITY ID REGISTER (PCICFG_VPD_CAP_ID, OFFSET 0x50)

Description	Mode	Reset	06	08	09	16
The Vital Product Data Capability ID register indicates that the next 8 bytes contain a RO Vital Product Data capability block. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440) " on page 214).	0x3		X	X	X	X

VPD NEXT CAPABILITY POINTER REGISTER (PCICFG_VPD_NEXT_CAP_PTR, OFFSET 0x51)

Description	Mode	Reset	06	08	09	16
The Next Capability register continues the PCI capability chain. Its value specifies an RO offset in the PCI address space of the next capability. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440) " on page 214).	0x58		X	X	X	X

VPD ADDRESS/FLAG REGISTER (PCICFG_VPD_FLAG_ADDR, OFFSET 0x52)

Table 67: VPD Address/Flag Register (pcicfg_vpd_flag_addr, Offset 0x52)

Bit	Name	Description	Mode	Reset	06	08	09	16
15	FLAG	This bit is used to control passing of data between the VPD_Data register and Non-Volatile memory. To read a value, this bit is written as 0 when the address is written. When the data is available to read, this bit will read as a one.	RW		X	X	X	X
14–2	ADDRESS	This value is the 32-bit word address of the VPD value being accessed in the VPD_Data register. Since the data register is 32-bits wide.	RW		X	X	X	X
1–0	RESERVED		RO	0	X	X	X	X

VPD DATA REGISTER (PCICFG_VPD_DATA, OFFSET 0x54)

Description	Mode	Reset	06	08	09	16
The VPD Data register has an undefined value at power-on reset. See the instructions for the FLAG bit above for usage of this register.	RW	0XXXX XXXXXX	X	X	X	X

MESSAGE SIGNALLED INTERRUPTS REGISTERS

MSI CAPABILITY ID REGISTER (PCICFG_MSI_CAP_ID, 0x58)

Description	Mode	Reset	06	08	09	16
The MSI Capability ID indicates that the next 8 bytes contain an MSI capability block. RO This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440)" on page 214).	0x5	X X X X				

MSI NEXT CAPABILITY POINTER REGISTER (PCICFG_MSI_NEXT_CAP_PTR, OFFSET 0x59)

Description	Mode	Reset	06	08	09	16
The Next Capability register continues the PCI capability chain. Its value specifies RO an offset in the PCI address space of the next capability. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440)" on page 214).	0	X X - -				
The Next Capability register continues the PCI capability chain. Its value specifies RO a offset in the PCI address space of the next capability. This value can be modified by firmware (see " PCI ID Value 4 Register (pci_id_val4, Offset 0x440)" on page 214).	0xa0	- - X X				

MSI CONTROL REGISTER (PCICFG_MSI_CONTROL, OFFSET 0x5A)

Table 68: MSI Control Register (pcicfg_msi_control, Offset 0x5a)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–8	RESERVED		RO	0	X	X	X	X
7	64_BIT_ADDR_CAP	This bit indicates that the chip is capable of generating 64 bit MSI messages.	RO	0x1	X	X	X	X
6–4	MENA	These bits indicate the number of message that the chip is configured (allowed) to generate.	RW	0	X	X	X	X
Value	Name	Description						
0	1	Chip is set to generate 1 message						
1	2	Chip is set to generate 2 messages						
2	4	Chip is set to generate 4 messages						
3	8	Chip is set to generate 8 messages						
4	16	Chip is set to generate 16 messages						
5	32	Chip is set to generate 32 messages						

Table 68: MSI Control Register (pcicfg_msi_control, Offset 0x5a) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
3-1	MCAP	These bits indicate the number of messages that the chip is capable of generating. This value comes from the missing			RO	0	X	X	X	X
		Value			Name	Description				
		0	1			Chip is capable of generating 1 message				
		1	2			Chip is capable of generating 2 messages				
		2	4			Chip is capable of generating 4 messages				
		3	8			Chip is capable of generating 8 messages				
		4	16			Chip is capable of generating 16 messages				
		5	32			Chip is capable of generating 32 messages				
0	ENABLE	When this bit is set, the chip will generate <u>MSI</u> cycles to indicate <u>interrupts</u> instead of asserting the INTA pin. When this bit is 0, the INTA pin will be used.			RW	0				

MSI ADDRESS LOW REGISTER (PCICFG_MSI_ADDR_L, OFFSET 0x5c)

Description	Mode	Reset	06	08	09	16
This register controls the lower half of the address of the MSI message that is generated. The lower 2 bits of this register should be programmed to 0 (32-bit aligned).		0xFFFF XXXXXX	X	X	X	X

MSI ADDRESS HIGH REGISTER (PCICIFG_MSI_ADDR_H, OFFSET 0x60)

Description	Mode	Reset	06	08	09	16
This register controls the upper half of the address of the MSI message that is generated.	RW	0xFFFF XXXX	X	X	X	X

MSI DATA REGISTER (PCICFG_MSI_DATA, OFFSET 0x64)

Description	Mode	Reset	06	08	09	16
This register controls the data value that will be presented on the lower 16 bits of the data bus during MSI messages. The MENA bit field from the "MSI Control Register (pcicfg_msi_control, Offset 0x5a)" on page 170 allows a specific number of the lower bits (up to 6) to be modified to indicate different interrupt conditions.	RW	0xFFFF XXXX	X	X	X	X

PRIVATE CONFIGURATION REGISTERS

MISCELLANEOUS CONFIGURATION REGISTER (PCICFG_MISC_CONFIG, OFFSET 0x68)

 **Note:** Unlike all other registers in PCI configuration space, the pcicfg_misc_config_register only supports 32-bit write operations. Partial writes to this register are not supported.

Table 69: Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	ASIC_ID	These bits are the hard coded ASIC type ID.	RO	0x6	X	–	–	–
31–28	ASIC_ID	These bits are the hard coded ASIC type ID.	RO	0x8	–	X	–	–
31–28	ASIC_ID	These bits are the hard coded ASIC type ID.	RO	0x9	–	–	X	X
27–24	ASIC_BASE_REV	These bits are the hard coded ASIC base revision (0 = A, 1 = B, and so on).	RO	0	X	X	X	X
23–16	ASIC_METAL_REV	These bits are the hard coded ASIC metal revision.	RO	0	X	X	X	X
15–10	RESERVED		RO	0	X	X	–	–
15–13	RESERVED		RO	0	–	–	X	X
12	GRC_WIN3_SWAP_EN	When this bit is set the swap settings TARGET_GRC_WORD_SWAP and TARGET_BYTE_SWAP are applied to GRC Window 3.	–	–	–	–	X	X
11	GRC_WIN2_SWAP_EN	When this bit is set the swap settings TARGET_GRC_WORD_SWAP and TARGET_BYTE_SWAP are applied to GRC Window 2.	–	–	–	–	X	X
10	GRC_WIN1_SWAP_EN	When this bit is set the swap settings TARGET_GRC_WORD_SWAP and TARGET_BYTE_SWAP are applied to GRC Window 1.	–	–	–	–	X	X
9–8	RESERVED		RO	0	–	–	X	X
9	CORE_RST_BSY	This bit reads as 1 after the CORE_RST_REQ bit self clears until the core reset is complete and the software may execute new target accesses to the chip.	RO	0	X	X	–	–
8	CORE_RST_REQ	When this bit is written as a 1, a CORE reset is generated. This bit will remain 1 until the reset request has been accepted by the core logic. The software must guarantee that it will not execute any target accesses after this bit is written as 1 until the CORE_RST_REQ and CORE_RST_BSY bits both read as 0.	SC	0	X	X	–	–

Table 69: Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7	REG_WINDOW_ENA	Setting this bit enables write access to the Reg_Window register. If this bit is 0, then the Reg_Window register access is disabled.	RW	0	X	X	X	X
6	TARGET_GRC_WORD_SWAP	Setting this bit enables word-swapping of target accesses of the GRC and Memory window area. This bit has no effect on configuration cycles or accesses to the Mailbox Queue area.	RW	0	X	X	X	X
5	CLOCK_CTL_ENA	Setting this bit enables write access to the Clock_Ctl register. If this bit is 0, then the Clock_Ctl register is read-only.	RW	0	X	X	-	-
4	PCI_STATE_ENA	Setting this bit enables write access to the PCI_State register. If this bit is 0, then the PCI_State register is read-only.	RW	0	X	X	-	-
5–4	RESERVED		RO	0	-	-	X	X
3	TARGET_MB_WORD_SWAP	Setting this bit enables word-swapping of target accesses of the mailbox space in the memory space mapped by the BAR_1 register. This bit has no effect on configuration cycles or accesses to the GRC, Memory window, and Mailbox Queue areas.	RW	0	X	X	X	X
2	TARGET_BYTE_SWAP	Setting this bit enables byte-swapping of all target accesses through the memory space mapped by the BAR_1 register. This bit has no effect on configuration cycles, but does effect accesses to the GRC, Memory window, and Mailbox Queue areas.	RW	0	X	X	X	X
1–0	RESERVED		RO	0	X	X	X	X

MISCELLANEOUS STATUS REGISTER (PCICFG_MISC_STATUS, OFFSET 0x6C)**Table 70: Miscellaneous Status Register (pcicfg_misc_status, Offset 0x6c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–9	RESERVED		RO	0	-	-	X	X
8	BAD_MEM_WRITE_BE	This bit is set on a partial DWORD write to any location other than the Mail Box Queue. This is not allowed since the internal GRC bus does not support partial DWORD transfers.	AC	0	-	-	X	X
7–1	RESERVED		RO	0	-	-	X	X
0	INTA_VALUE	This bit indicates the level of the external INTA pin. This is the actual value on the PCI bus read back through the pin input buffer and is used for debug purposes.	RO	1	-	-	X	X
31–6	RESERVED		RO	0	X	X	-	-

Table 70: Miscellaneous Status Register (pcicfg_misc_status, Offset 0x6c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16	
5–4	PCIX_SPEED	These bits indicate the clock speed mode that was captured from the PCIX bus at reset.	RO	0	X	X	–	–	
			Value	Name	Description				
				0	66	Bus Indicates 66 MHz			
				1	100	Bus Indicates 100 MHz			
				2	133	Bus Indicates 133 MHz			
				3	PCI_MODE	Bus Indicates PCI Mode			
3	PCIX_DET	This bit is set when the PCI interface detects that it is in PCIX system.	RO	0	X	X	–	–	
2	M66EN	This bit indicates the current state of the M66EN input pin. In PCIX mode, this pin is indeterminate.	RO	0	X	X	–	–	
1	32BIT_DET	This bit is set when the PCI interface detects that it is in a 32-bit slot. This means that the REQ64 signal was high at reset.	RO	0	X	X	–	–	
0	INTA_VALUE	This bit indicates the level of the external INTA pin. This is the actual value on the PCI bus read back through the pin input buffer and is used for debug purposes.	RO	0	X	X	–	–	

PCI CLOCK CONTROL REGISTER (PCICFG_PCI_CLOCK_CONTROL_BITS, OFFSET 0x70)

This register is the PCI clock domain control register for clock control. There is a separate CORE domain control register in the MISC block (see “[MSI Control Register \(pcicfg_msi_control, Offset 0x5a\)](#)” on page 170). The values of these two registers are logically ORed together to create the actual clock control register value. The read RO bit are valid in both registers. This register is reset by a CORE reset.

Table 71: PCI Clock Control Register (pcicfg_pci_clock_control_bits, Offset 0x70)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	–	–	X	X
31–20	RESERVED		RW	0	X	X	–	–
19	RESERVED		RW	0	X	X	–	–
18	RESERVED		RW	0	X	X	–	–
17	PCI_PLL_STOP	This bit powers down the PCI PLL when it is set. This is only used for testing.	RW	0	X	–	–	–
17	RESERVED		RW	0	–	X	–	–
16	CORE_CLK_PLL_STOP	This bit stops the CORE_CLK PLL when it is set. One of the alternate clocks must be selected when this is done. (ALT = 1)	RW	0	X	X	–	–

Table 71: PCI Clock Control Register (pcicfg_pci_clock_control_bits, Offset 0x70) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16																	
15–12	CORE_CLK_PLL_SPEED	These bits control the core clock PLL settings to select different core clock speeds. The clock generator will pick the slowest of the speeds enabled. If no bits are set, then the PLL will run at 100-MHz core clock speed. These bits must only be modified when the ALT bit is 1. The Firmware/software must guarantee that the ALT bit changes and the PLL_SPEED changes are not in the same PCI or GRC cycle. The Hardware must only modify the PLL output divisor with this value and not the PLL feedback programming. At reset, this value is to 0x2 if the VAUX_PRESENT bit input is 1 and the PCI Pad power detect indicates no PCI pad power. Otherwise, this bit resets to 0x0.	RW	0	X	X	–	–																	
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>100</td><td>100 MHz (PLL Divisor == 4)</td></tr> <tr> <td>1</td><td>80</td><td>80 MHz (PLL Divisor == 5)</td></tr> <tr> <td>2</td><td>50</td><td>50 MHz (PLL Divisor == 8)</td></tr> <tr> <td>4</td><td>40</td><td>40 MHz (PLL Divisor == 10)</td></tr> <tr> <td>8</td><td>25</td><td>25 MHz (PLL Divisor == 16)</td></tr> </tbody> </table>	Value	Name	Description	0	100	100 MHz (PLL Divisor == 4)	1	80	80 MHz (PLL Divisor == 5)	2	50	50 MHz (PLL Divisor == 8)	4	40	40 MHz (PLL Divisor == 10)	8	25	25 MHz (PLL Divisor == 16)					
Value	Name	Description																							
0	100	100 MHz (PLL Divisor == 4)																							
1	80	80 MHz (PLL Divisor == 5)																							
2	50	50 MHz (PLL Divisor == 8)																							
4	40	40 MHz (PLL Divisor == 10)																							
8	25	25 MHz (PLL Divisor == 16)																							
11	PLAY_DEAD	This bit disables all CORE_CLK and CPU_CLK for blocks and is used to place the chip into play-dead mode used to minimize VAUX power usage when not configured for VAUX PME assertion. Setting this bit in the MISC block is not a good idea as the PCI cycle may not finish. This bit also sets IDQ mode for all devices in the chip other than the CK25 oscillator, 1.2V regulator, PME pad and GPIO2 pad.	RW	0	X	X	–	–																	
10–8	CORE_CLK_ALT_SRC	These bits select the alternate clock source. The clock generator will pick the slowest of the speeds enabled. These bits must only be modified when the ALT bit is 0.	RW	0	X	X	–	–																	
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>UNDEF</td><td>Undefined alternate clock select.</td></tr> <tr> <td>1</td><td>12</td><td>Select 25-MHz Reference Clock to make a 12.5-MHz core clock.</td></tr> <tr> <td>2</td><td>6</td><td>Select 12.5-MHz Clock to make a 6.25-MHz core clock.</td></tr> <tr> <td>4</td><td>62</td><td>Select 125-MHz PHY Clock to make a 62.5-MHz core clock.</td></tr> </tbody> </table>	Value	Name	Description	0	UNDEF	Undefined alternate clock select.	1	12	Select 25-MHz Reference Clock to make a 12.5-MHz core clock.	2	6	Select 12.5-MHz Clock to make a 6.25-MHz core clock.	4	62	Select 125-MHz PHY Clock to make a 62.5-MHz core clock.								
Value	Name	Description																							
0	UNDEF	Undefined alternate clock select.																							
1	12	Select 25-MHz Reference Clock to make a 12.5-MHz core clock.																							
2	6	Select 12.5-MHz Clock to make a 6.25-MHz core clock.																							
4	62	Select 125-MHz PHY Clock to make a 62.5-MHz core clock.																							
7	CORE_CLK_ALT	This bit selects the alternate CORE_CLK source. This is used to select a non-PLL source while the PLL is being manipulated or to select a non-PLL source during WOL mode so the PLL can be powered down all together.	RW	0	X	X	–	–																	
6	CORE_CLK_DISABLE	This bit disables CORE_CLK and CPU_CLK to all blocks that are not needed for WOL operation. At reset, this bit is to 1 if the VAUX_PRESENT bit input is 1 and the PCI Pad power detect indicates no PCI pad power for the PCI version of this register only. Otherwise, this bit resets to 0. The MISC block version of this register always resets this bit to 0.	RW	X	X	X	–	–																	
5–4	RESERVED		RO	0	X	X	–	–																	

Table 71: PCI Clock Control Register (pcicfg_pci_clock_control_bits, Offset 0x70) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
3–0	PCI_CLK_SPD_DET	The values indicates the detected speed of the PCI clock as compared to the 25-MHz crystal clock.			RO	0	X	X	—	—
	Value	Name	Description							
	0	32MHz	PCI clock is in the range 28.1 to 39.1 MHz							
	1	38MHz	PCI clock is in the range 34.3 to 45.3 MHz							
	2	48MHz	PCI clock is in the range 40.6 to 41.6 MHz							
	3	55MHz	PCI clock is in the range 46.8 to 60.9 MHz							
	4	66MHz	PCI clock is in the range 56.2 to 71.9 MHz							
	5	80MHz	PCI clock is in the range 67.1 to 90.5 MHz							
	6	95MHz	PCI clock is in the range 85.9 to 103 MHz							
	7	133MHz	PCI clock is in the range 98.4 to 198 MHz							
	f	LOW	PCI clock is in the range 0 to 28.2 MHz							

REGISTER WINDOW ADDRESS REGISTER (PCICFG_REG_WINDOW_ADDRESS, OFFSET 0x78)

Description	Mode	Reset	06	08	09	16
This register is the address register used to access the internal register bus from PCI configuration space. All registers that can be accessed in memory-mapped mode can be accessed using this address. This method of access will work even when the memory space is mapped.	RW	0	X	X	X	X
When using the Register Window Address and Register Window Data registers to perform indirect register accesses, Broadcom recommends that the host software use PCI configuration cycles rather than memory mapped I/O cycles to guarantee that the Address register is completely written to the controller before the Data register is accessed. If memory-mapped I/O cycles are used, then every register write to the Address and Data register must be followed by a read from the same register to guarantee that any posted writes buffers are flushed to the NetXtreme II controller.						

REGISTER WINDOW DATA REGISTER (PCICFG_REG_WINDOW, OFFSET 0x80)

Description	Mode	Reset	06	08	09	16
This register is the data window register used to access the internal register bus from PCI configuration space. All registers that can be accessed in memory-mapped mode can be accessed through this window instead. This window will work even when the memory space is mapped. When this register is read, the data will be read from the internal register bus and passed to this register before the transfer is complete. This data register, like the rest of the configuration during configuration cycles will not be subjected to byte or word-swapping. This means that the address specified is a true 32-bit address into the GRC register space.	RW	0	X	X	X	X

INTERRUPT ACKNOWLEDGE COMMAND REGISTER (PCICFG_INT_ACK_CMD, OFFSET 0x84)

The value of this register is used by the Host Coalescing (HC) block to handle the generation of the interrupt signal. The value may be read back through this register, but that value will always be the last value written.



Note: This register is cleared by CORE reset so that it remains synchronized with the Host Coalescing logic.

Table 72: Interrupt Acknowledge Command Register (pcicfg_int_ack_cmd, Offset 0x84)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	–	–	X	X
31–19	RESERVED		RO	0	X	X	–	–
27–24	INTERRUPT_NUM	This field is used to indicate which status block is being acknowledged.		0	–	–	X	X
23–19	RESERVED		RO	0	–	–	X	X
18	MASK_INT	When this bit is set, the INTA output is masked and will not drive active. This bit should be set inside the Hardware Interrupt Handler routine called by the operating system to mask the generation of more interrupts until the current interrupts can be handled.	RW	0x1	X	X	X	X
17	USE_INT_HC_PARAM	When this bit is set, the Host Coalescing block will use the During Interrupt settings rather than the normal settings for generating new status blocks (and subsequent interrupts).	RW	0	X	X	X	X
16	INDEX_VALID	When this bit is set, the Host Coalescing block will use the INDEX field to acknowledge status blocks that have been generated. If this bit is not set on a write to this register, then the INDEX field will be ignored and interrupts will not be acknowledged.	RW	0	X	X	X	X
15–0	INDEX	After an interrupt is handled, the status_idx field of the "Host Status Block (Status)" on page 73 (read before processing any values from the block) must be written to this field. This is used by the Host Coalescing block to re-interrupt if the latest block has not been processed.	RW	0	X	X	X	X

STATUS BIT SET COMMAND REGISTER (PCICFG_STATUS_BIT_SET_CMD, OFFSET 0x88)

Description	Mode	Reset	06	08	09	16
This register always returns a value of 0x0 when read. Each bit written as a 1 in this RW register will cause the corresponding bit in the Status Block (see " Host Status Block Attention Acknowledge (status_attn_bits_ack) " on page 77) to be set the next time the block is generated. This will acknowledge that the driver has recognized that the bit position in the status_bits value of the current Status Block as being 1.	0	X X X X				

STATUS BIT CLEAR COMMAND REGISTER (PCICFG_STATUS_BIT_CLEAR_CMD, OFFSET 0x8C)

Description	Mode	Reset	06	08	09	16
This register always returns a value of 0x0 when read. Each bit written as a 1 in this RW register will cause the corresponding bit in the Status Block (see " Host Status Block Attention Acknowledge (status_attn_bits_ack) " on page 77) to be cleared the next time the block is generated. This will acknowledge that the driver has recognized that the bit position in the status_bits value of the current Status Block as being 0.	0	X X X X				

MAILBOX QUEUE ADDRESS REGISTER (PCICFG_MAILBOX_QUEUE_ADDR, OFFSET 0x90)

Description	Mode	Reset	06	08	09	16
This register controls where, in the Mailbox Queue Address space, writes to the " Mailbox Queue Data Register (pcicfg_mailbox_queue_data, Offset 0x94) " on page 178 will be placed. This interface is only for kernel drivers that must control the NetXtreme II completely through PCI configuration space. The address value includes offset 0x10000 where mailboxes start when using PCI memory cycles.	RW	0	X X X X			

MAILBOX QUEUE DATA REGISTER (PCICFG_MAILBOX_QUEUE_DATA, OFFSET 0x94)

Description	Mode	Reset	06	08	09	16
When the write-only register is written, the value will be passed to the Mailbox Queue block as a write performed to the offset specified in the " Mailbox Queue Address Register (pcicfg_mailbox_queue_addr, Offset 0x90) " on page 178. Reads from this register will always return 0. This data register is not subject to byte/word-swapping.	WO	0	X X X X			

Note: The HALT_DIS bit of the "[MQ Configuration Register \(mq_config, Offset 0x3c08\)](#)" on page 405 should be set for BCM5709 AX controllers to prevent OS initiated reads from this register from setting the RD_ERROR bit of the "[MQ Command Register \(mq_command, Offset 0x3c00\)](#)" on page 403.

MSI-X CAPABILITIES REGISTERS

MSI-X CAPABILITY ID REGISTER (PCICFG_MSIX_CAP_ID, OFFSET 0xA0)

Description	Mode	Reset	06	08	09	16
The MSI-X Capability ID indicates the beginning of the MSI-X capability block.	RO	0x11	-	-	X	X

MSI-X NEXT CAPABILITY POINTER REGISTER (PCICFG_MSIX_NEXT_CAP_PTR, OFFSET 0xA1)

Description	Mode	Reset	06	08	09	16
The Next Capability register continues the PCI capability chain. Its value specifies RO the offset in the PCI configuration address space of the next capability.		0xac	-	-	X	X

MSI-X CONTROL REGISTER (PCICFG_MSIX_CONTROL, OFFSET 0xA2)

Table 73: MSI-X Control Register (pcicfg_msix_control, Offset 0xa2)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–0	RESERVED		RO	0	X	X	–	–
15	MSIX_ENABLE	When this bit is set, and the MSI enable bit in the MSI message control register is 0, the function is permitted to use MSIX request service and prohibited from using INTx# messages.	RW	0	–	–	X	X
14	FUNC_MASK	When this bit is set, all of the vectors associated with the function are masked regardless of their per vector Mask bit.	RW	0	–	–	X	X
13–11	RESERVED		RO	0	–	–	X	X
10–0	TABLE_SIZE	This field is read by system software to determine the MSI-X table size N, which is encoded as N-1.	RO	0	–	–	X	X

This value can be modified by firmware.

(Unresolved for PG303)

(Unresolved for PG203)

MSI-X TABLE OFFSET BASE INDEX REGISTER (PCICFG_MSIX_TBL_OFF_BIR, OFFSET 0xA4)*Table 74: MSI-X Table Offset Base Register (pcicfg_msix_tbl_off_bir, Offset 0xa4)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–3	TABLE_OFFSET	–	RO	0	–	–	X	X
2–0	TABLE_BIR	This indicates which function's BAR is used to map MSI-X table into memory space.	RO	0	–	–	X	X

This value can be modified by firmware.

(Unresolved for PG303)

(Unresolved for PG203)

MSI-X PBA OFFSET BASE REGISTER (PCICFG_MSIX_PBA_OFF_BIR, OFFSET 0xA8)*Table 75: MSI-X PBA Offset Base Register (pcicfg_msix_pba_off_bir, Offset 0xa8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–3	PBA_OFFSET	–	RO	0	–	–	X	X
2–0	PBA_BIR	This indicates which function's BAR is used to map MSI-X PBA into memory space.	RO	0	–	–	X	X

This value can be modified by firmware.

(Unresolved for PG303)

(Unresolved for PG203)

PCI EXPRESS CAPABILITIES REGISTERS

PCI EXPRESS CAPABILITY ID REGISTER (PCICFG_PCIE_CAP_ID, OFFSET 0XAC)

Description	Mode	Reset	06	08	09	16
The PCI Express Capability ID indicates the beginning of the PCI Express RO capabilities block.		0x10	-	-	X	X

PCI EXPRESS NEXT CAPABILITY POINTER REGISTER (PCICFG_PCIE_NEXT_CAP_PTR, OFFSET 0XAD)

Description	Mode	Reset	06	08	09	16
The Next Capability register continues the PCI capability chain. Its value specifies RO the offset in the PCI configuration address space of the next capability.	0		-	-	X	X

PCI EXPRESS CAPABILITIES REGISTER (PCICFG_PCIE_CAPABILITY, OFFSET 0XAE)

Table 76: PCI Express Capabilities Register (pcicfg_PCIE_capability, Offset 0xae)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–8	RESERVED		RO	0	X	X	X	X
7–0	RESERVED		RO	0	X	X	-	-
7–4	TYPE	Device/Port Type. EndPoint	RO	0	-	-	X	X
3–0	VER	Capability Version. PCI Express Capability structure version number	RO	0x2	-	-	X	X

This value can be modified by firmware.

(Unresolved for PG303)

(Unresolved for PG203)

PCI EXPRESS DEVICE CAPABILITIES REGISTER (PCICFG_DEVICE_CAPABILITY, OFFSET 0XB0)

Table 77: PCI Express Device Capabilities Register (pcicfg_device_capability, Offset 0xb0)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	RESERVED		RO	0	X	X	-	-
15	ROLE_BASED_ERR_RPT	Role-based error reporting.	RO	1	-	-	X	X
14–0	RESERVED		RO	0	X	X	-	-
14–12	RESERVED		RO	0	-	-	X	X



Table 77: PCI Express Device Capabilities Register (pcicfg_device_capability, Offset 0xb0) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11–9	EP_L1_ACCP_LAT	Endpoint L1 Acceptable Latency	RO	0	—	—	X	X
8–6	EP_LOS_ACCP_LAT	Endpoint L0s Acceptable Latency	RO	0	—	—	X	X
5	EXT_TAG_SUPPT	Extended Tag Field Support	RO	0	—	—	X	X
4–3	PHANTOM_SUPPT	Phantom Functions Supported	RO	0	—	—	X	X
2–0	MAX_PAYLOAD	Max Payload Size Supported	RO	0x2	—	—	X	X

This value can be modified by firmware (see “PCI Express Device Control 2 Register (pccfg_device_control_2, Offset 0xd4)” on page 187).

PCI EXPRESS DEVICE CONTROL REGISTER (PCICFG_DEVICE_CONTROL, OFFSET 0XB4)

Table 78: PCI Express Device Control Register (pcicfg_device_control, Offset 0xb4)

Bit	Name	Description	Mode	Reset	06	08	09	16
15	RESERVED		RO	0	X	X	X	X
14–0	RESERVED		RO	0	X	X	—	—
14–12	MAX_RD_REQ	Maximum Read Request Size	RW	0x2	—	—	X	X
11	ENA_NO_SNOOP	Enable No Snoop. When this bit is set to 1, PCIE initiates a read request with the No Snoop bit in the attribute field set for the transactions that request the No Snoop attribute.	RW	0x1	—	—	X	X
10	AUX_PWR_PM_ENA	This bit when set enables device to draw aux power independent of PME AUX power	RW	0x1:Hard	—	—	X	X
9	RESERVED		RO	0	—	—	X	X
8	EXT_TAG_ENA	Extended Tag Field Enable	RW	0	—	—	X	X
7–5	MAX_PAYLOAD	Max Payload Size	RW	0	—	—	X	X
4	RELAX_ENA	Relax Ordering Enable	RW	0x1	—	—	X	X
3	UNSUP_REQ_ENA	Unsupported Request Reporting Enable	RW	0	—	—	X	X
2	FATAL REP ENA	Fatal Error Reporting Enable	RW	0	—	—	X	X
1	NON_FATAL REP ENA	Non-Fatal Error Reporting Enable	RW	0	—	—	X	X
0	CORR_ERR REP ENA	Correctable Error Reporting Enable	RW	0	—	—	X	X

PCI EXPRESS DEVICE STATUS REGISTER (PCICFG_DEVICE_STATUS, OFFSET 0XB6)*Table 79: PCI Express Device Status Register (pcicfg_device_status, Offset 0xb6)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
15–6	RESERVED		RO	0	—	—	X	X
5–0	RESERVED		RO	0	X	X	—	—
5	NO_PEND	This bit will return a 1 whenever a non-posted request initiated by PCIE core is pending to be completed.	RO	0	—	—	X	X
4	AUX_PWR_DET	This bit returns the current state of the VAUX_PRSNT pin of the device. When it is '1', it is indicating that part needs VAUX and detects the VAUX is present.	RO	0	—	—	X	X
3	UNSUP_REQ_DET	UnSupported Request Detected	WC	0	—	—	X	X
2	FATAL_ERR_DET	Fatal Error Detected	WC	0	—	—	X	X
1	NON_FATAL_ERR_DET	Non-Fatal Error Detected	WC	0	—	—	X	X
0	CORR_ERR_DET	Correctable Error Detected	WC	0	—	—	X	X

PCI EXPRESS LINK CAPABILITY REGISTER (PCICFG_LINK_CAPABILITY, OFFSET 0XB8)

Table 80: PCI Express Link Capability Register (pcicfg_link_capability, Offset 0xb8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–24	PORT_NUMBER	PCIE Port Number	RO	0	–	–	X	X
23–19	RESERVED		RO	0	–	–	X	X
18	CLK_PWR_MGMT	Clock Power Management	RO	0	–	–	X	X
17–15	L1_EXIT_LAT	L1 Exit Latency.	RO	0x2	–	–	X	X
		<i>Value</i> <i>Name</i> <i>Description</i>						
		1 1_2 L1 exit latency of 1 µs to 2 µs.						
		2 2_4 L1 exit latency of 2 µs to 4 µs.						
14–12	L0S_EXIT_LAT	L0s Exit Latency.	RO	0x6	–	–	X	X
		<i>Value</i> <i>Name</i> <i>Description</i>						
		5 1_2 L0s exit latency of 1 µs to 2 µs.						
		6 2_4 L0s exit latency of 2 µs to 4 µs.						
11–10	ASPM_SUPT	ASPM Support.	RO	0x3	–	–	X	X
		<i>Value</i> <i>Name</i> <i>Description</i>						
		0 RES_0 Reserved						
		0 L0S L0s entry supported						
		0 RES_2 Reserved						
		0 L0S_L1 L0s and L1 supported						
9–4	MAX_LINK_WIDTH	Maximum Link Width.	RO	0x4	–	–	X	X
		<i>Value</i> <i>Name</i> <i>Description</i>						
		1 1 One Lane Max.						
		2 2 Two Lanes Max.						
		4 4 Four Lanes Max.						
		8 8 Eight Lanes Max.						
3–0	MAX_LINK_SPEED	Value used by internal logic is the smaller of the value programmed for each function.	RO	0x3	–	–	X	X
		<i>Value</i> <i>Name</i> <i>Description</i>						
		1 2_5 2.5 Gbps Max.						
		2 5 5 Gbps Max.						

This value can be modified by firmware.

(Unresolved for PG303) (Unresolved for PG203)



PCI EXPRESS LINK CONTROL REGISTER (PCICFG_LINK_CONTROL, OFFSET 0xbc)*Table 81: PCI Express Link Control Register (pcicfg_link_control, Offset 0xbc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
15–8	RESERVED		RO	0	—	—	X	X
7–0	RESERVED		RO	0	X	X	—	—
7	LINK_CR_EXT_SYNC	Extended Synch. This bit when set forces the transmission of 4096 FTS ordered sets in the L0s state followed by a single SKP ordered set prior to entering the L0 state, and the transmission of 1024 TS1 ordered sets in the L1 state prior to entering the Recovery state. Value used by logic is resolved to 1 if either function has this bit set.	RW	0	—	—	X	X
6	LINK_CR_COMMON_CLK	Common Clock Configuration. Value used by logic is resolved to 1 only if both functions (when enabled) have this bit set.	RW	0	—	—	X	X
5–4	RESERVED		RO	0	—	—	X	X
3	RD_COMP_BOUND	Read Completion Boundary.	RW	0	—	—	X	X
		Value	Name	Description				
		0	64	64 bytes				
		1	128	128 bytes				
2	RESERVED		RO	0	—	—	X	X
1–0	ASPM_CTRL	ASPM Control. Value used by logic is dependent on the value of this bit for each enabled function and also on the programmed power state of each function.	RW	0	—	—	X	X

PCI EXPRESS LINK STATUS REGISTER (PCICFG_LINK_STATUS, OFFSET 0xbe)*Table 82: PCI Express Link Status Register (pcicfg_link_status, Offset 0xbe)*

Bit	Name	Description	Mode	Reset	06	08	09	16
15–13	RESERVED		RO	0	—	—	X	X
12–0	RESERVED		RO	0	X	X	—	—
12	SLOT_CLK	Slot Clock configuration.	RO	0	—	—	X	X
11	TRAINING	Link Training.	RO	0	—	—	X	X
10	TRAINING_ERR	Training Error.	RO	0	—	—	X	X
9–4	NEG_LINK_WIDTH	Negotiated Link Width. These bits indicate the negotiated link width of the PCI Express link.	RO	0	—	—	X	X
3–0	SPEED	Link Speed. These bits indicate the negotiated link speed of the PCI Express link.	RO	0	—	—	X	X

PCI EXPRESS SLOT CAPABILITY REGISTER (PCICFG_SLOT_CAPABILITY, OFFSET 0xC0)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset and is not supported. RW	0	- - X X				

PCI EXPRESS SLOT CONTROL REGISTER (PCICFG_SLOT_CONTROL, OFFSET 0xC4)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset and is not supported. RW	0	- - X X				

PCI EXPRESS SLOT STATUS REGISTER (PCICFG_SLOT_STATUS, OFFSET 0xC6)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset and is not supported.	RW	0 - - X X				

PCI EXPRESS ROOT CONTROL REGISTER (PCICFG_ROOT_CONTROL, OFFSET 0xC8)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset and is not supported.	RW	0 - - X X				

PCI EXPRESS ROOT CAPABILITIES REGISTER (PCICFG_ROOT_CAP, OFFSET 0xCA)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset and is not supported.	RW	0 - - X X				

PCI EXPRESS ROOT STATUS REGISTER (PCICFG_ROOT_STATUS, OFFSET 0xCC)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset and is not supported.	RW	0 - - X X				

PCI EXPRESS DEVICE CAPABILITY 2 REGISTER (PCICFG_DEVICE_CAPABILITY_2, OFFSET 0xD0)

Table 83: PCI Express Device Capability 2 Register (pcicfg_device_capability_2, Offset 0xd0)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–5	RESERVED				RO	0	X	X
4–0	RESERVED				RO	0	X	X
4	CMPL_TO_DISABL_SUPP	Completion timeout disable supported	RO	0x1	–	–	X	X
3–0	CMPL_TO_RANGE_SUPP	Completion timeout ranges supported	RO	0xf	–	–	X	X
			Value	Name	Description			
			15	ABCD	Ranges A, B, C, and D			

This value can be modified by firmware (see “PCI PCIe Device Capability 2 Register (pci_pcie_device_capability_2, offset 0x4e4)” on page 372).

(Unresolved for PG303)

(Unresolved for PG203)

PCI EXPRESS DEVICE CONTROL 2 REGISTER (PCCFG_DEVICE_CONTROL_2, OFFSET 0xD4)

Table 84: PCI Express Device Control 2 Register (pccfg_device_control_2, Offset 0xd4)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–5	RESERVED				RO	0	–	X
4–0	RESERVED				RO	0	X	X
4	CMPL_TO_DISABLE	Completion timeout disable	RW	0	–	–	X	X
3–0	CMPL_TO_VALUE	Completion timeout value	RW	0	–	–	X	X
			Value	Name	Description			
			0	50MS	50 ms			
			1	100US	100 µs			
			2	10MS	10 ms			
			3	55MS	55 ms			
			4	210MS	210 ms			
			5	900MS	900 ms			
			6	3_5S	3.5s			
			7	13S	13s			
			8	64S	64s			

PCI EXPRESS DEVICE STATUS 2 REGISTER (PCICFG_DEVICE_STATUS_2, OFFSET 0xD6)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset.	RW	0	-	-	X	X

PCI EXPRESS LINK CAPABILITY 2 REGISTER (PCICFG_LINK_CAPABILITY_2, OFFSET 0xD8)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This read-write register has a value of 0 after power-on reset.	RW	0	-	-	X	X
This value can be modified by firmware.						

PCI EXPRESS LINK CONTROL 2 REGISTER (PCICFG_LINK_CONTROL_2, OFFSET 0xDC)

This register will be Read only by default, and will read all 0's to allow compliance with PCIe specification v1.1. To enable this register, set the COMPLY_PCIE_1_1 bit of the “PCI PCIe Capability Register (pci_pcie_capability, offset 0x4d0)” on page 368.

(Unresolved for PG303)

(Unresolved for PG203)

Table 85: PCI Express Link Control 2 Register (pcicfg_link_control_2, Offset 0xdc)

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
15–10	RESERVED		RO	0	X	X	X	X
9–0	RESERVED		RO	0	X	X	-	-

Table 85: PCI Express Link Control 2 Register (pcicfg_link_control_2, Offset 0xdC) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
9-7	TX_MARGIN	Controls the value of non de-emphasized voltage level at the TX pins. Value used by logic is resolved to the smaller binary value, if two functions have different values.	RW	0:Hard	-	-	X	X
Value Name								
0	000	800 mV–1200 mV for full swing and 400 mV–600 mV for half swing.						
1	001	Values will be monotonic with non zero Slope.						
2	010	Values will be monotonic with non zero Slope.						
3	011	200 mV–400 mV for full swing and 100 mV–200 mV for half swing						
4	100	Reserved						
5	101	Reserved						
6	110	Reserved						
7	111	Reserved						
6	SEL_DEEMPHASIS	When link is operating at Gen2 rates, this bit selects the level of de-emphasis. Value used by logic is resolved to 1 if either function has this bit set.	RW	0:Hard	-	-	X	X
Value Name								
0	0	-6 dB						
1	1	-3.5 dB						
5	HW_AUTO_SPEED_DISABLE	Not Supported	RO	0	-	-	X	X
4	ENTER_COMPLIANCE	S/W instructs link to enter compliance mode. Value used by internal logic is set when either function has this bit enabled.	RW	0:Hard	-	-	X	X
3-0	TARGET_LINK_SPEED	Upper limit of link speed	RW	0x1	-	-	X	X
Value Name								
0	2_5	2.5 Gbps						
1	5_0	5.0 Gbps						

PCI EXPRESS LINK STATUS 2 REGISTER (PCICFG_LINK_STATUS_2, OFFSET 0xDE)**Table 86: PCI Express Link Status 2 Register (pcicfg_link_status_2, Offset 0xde)**

Bit	Name	Description	Mode	Reset	06	08	09	16
15-0	RESERVED		RO	0	X	X	-	-

Table 86: PCI Express Link Status 2 Register (pcicfg_link_status_2, Offset 0xde) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–1	RESERVED		RO	0	–	–	X	X
0	DEEMPHASIS	Current de-emphasis value.	RO	0	–	–	X	X
Value Name Description								
0	6DB	-6 dB						
1	3_5DB	-3.5 dB						

PCI EXPRESS DEVICE SERIAL NUMBER CAPABILITIES REGISTERS

PCI EXPRESS DEVICE SERIAL NUMBER CAPABILITY ID REGISTER (PCICFG_DEVICE_SER_NUM_CAP_ID, OFFSET 0x100)

Description	Mode	Reset	06	08	09	16
This register indicates the beginning of the Device Serial Number Capability RO block. This value can be modified by firmware.	0x3	- - X X				

PCI EXPRESS DEVICE SERIAL NUMBER NEXT CAPABILITY POINTER REGISTER (PCICFG_DEVICE_SER_NUM_CAP_OFF, OFFSET 0x102)

Table 87: PCI Express Device Serial Number Next Capability Pointer Register (pcicfg_device_ser_num_cap_off, Offset 0x102)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–4	NEXT	Next Capabilities Pointer.	RO	0x110	- - X X			
3–0	VER	Capability ID Version.	RO	0x1	- - X X			

PCI EXPRESS LOWER SERIAL NUMBER REGISTER (PCICFG_LOWER_SER_NUM, OFFSET 0x104)

Description	Mode	Reset	06	08	09	16
The Lower Serial Number register returns the PCIe Device Serial Number bits RO [31:0]. This value can be modified by firmware).	0	- - X X				

PCI EXPRESS UPPER SERIAL NUMBER REGISTER (PCICFG_UPPER_SER_NUM, OFFSET 0x108)

Description	Mode	Reset	06	08	09	16
The Upper Serial Number register returns the PCIe Device Serial Number bits RO [63:32]. This value can be modified by firmware.	0	- - X X				

PCI EXPRESS ADVANCED ERROR CAPABILITIES REGISTERS

PCI EXPRESS ADVANCED ERROR CAPABILITY ID REGISTER (PCICFG_ADV_ERR_CAP_ID, OFFSET 0x110)

Description	Mode	Reset	06	08	09	16
The Advanced Error Capability ID indicates the beginning of the Advanced Error Capability block.	RO	0x1	-	-	X	X

PCI EXPRESS ADVANCED ERROR NEXT CAPABILITY POINTER REGISTER (PCICFG_ADV_ERR_CAP_OFF, OFFSET 0x112)

Table 88: PCI Express Advanced Error Next Capability Pointer Register (pcicfg_adv_err_cap_off, Offset 0x112)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–4	NEXT	Next Capabilities Pointer	RO	0x150	-	-	X	X
3–0	VER	Capability ID Version	RO	0x1	-	-	X	X

PCI EXPRESS UNCORRECTABLE ERROR STATUS REGISTER (PCICFG_UCORR_ERR_STATUS, OFFSET 0x114)

Table 89: PCI Express Uncorrectable Error Status Register (pcicfg_ucorr_err_status, Offset 0x114)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	-	-	X	X
20	URES	Unsupported Request Error Status	WC	0: Hard	-	-	X	X
19	ECRCS	ECRC Error Status	WC	0: Hard	-	-	X	X
18	MTLPS	Malformed TLP Status	WC	0: Hard	-	-	X	X
17	ROS	Receiver Overflow Status	WC	0: Hard	-	-	X	X
16	UCS	Unexpected Completion Status	WC	0: Hard	-	-	X	X
15	CAS	Completer Abort Status	WC	0: Hard	-	-	X	X
14	CTS	Completer Timeout Status	WC	0: Hard	-	-	X	X
13	FCPES	Flow Control Protocol Error Status	WC	0: Hard	-	-	X	X
12	PTLPS	Poisoned TLP Status	WC	0: Hard	-	-	X	X
11–5	RESERVED		RO	0	-	-	X	X
4	DLPES	Data Link Protocol Error Status	WC	0: Hard	-	-	X	X
3–0	RESERVED		RO	0	-	-	X	X

PCI EXPRESS UNCORRECTABLE ERROR MASK REGISTER (PCICFG_UCORR_ERR_MASK, OFFSET 0x118)

Table 90: PCI Express Uncorrectable Error Mask Register (pcicfg_ucorr_err_mask, Offset 0x118)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	–	–	X	X
20	UREM	Unsupported Request Error Mask	RW	0:Hard	–	–	X	X
19	ECRCM	ECRC Error Mask	RW	0:Hard	–	–	X	X
18	MTLPM	Malformed TLP Mask	RW	0	–	–	X	X
17	ROM	Receiver Overflow Mask	RW	0:Hard	–	–	X	X
16	UCM	Unexpected Completion Mask	RW	0:Hard	–	–	X	X
15	CAM	Completer Abort Mask	RW	0:Hard	–	–	X	X
14	CTM	Completer Timeout Mask	RW	0:Hard	–	–	X	X
13	FCPEM	Flow Control Protocol Error Mask	RW	0:Hard	–	–	X	X
12	PTLPM	Poisoned TLP Mask	RW	0:Hard	–	–	X	X
11–6	RESERVED		RO	0	–	–	X	X
5	SDEM	Surprise Down Error Mask	RO	0:Hard	–	–	X	X
4	DLPEM	Data Link Protocol Error Mask	RW	0:Hard	–	–	X	X
3–0	RESERVED		RO	0	–	–	X	X

PCI EXPRESS UNCORRECTABLE ERROR SEVERITY REGISTER (PCICFG_UCORR_ERR_SEVR, OFFSET 0x11c)

Table 91: PCI Express Uncorrectable Error Severity Register (pcicfg_ucorr_err_sevr, Offset 0x11c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	–	–	X	X
20	URES	Unsupported Request Error Severity	RW	0:Hard	–	–	X	X
19	ECRCES	Ecrc error Severity	RW	0:Hard	–	–	X	X
18	MTLPS	Malformed TLP Severity	RW	0x1:Hard	–	–	X	X
17	ROS	Receiver Overflow Severity	RW	0x1:Hard	–	–	X	X
16	UCS	Unexpected Completion Severity	RW	0:Hard	–	–	X	X
15	CAS	Completer Abort Severity	RW	0:Hard	–	–	X	X
14	CTS	Completer Timeout Severity	RW	0:Hard	–	–	X	X
13	FCPES	Flow Control Protocol Error Severity	RW	0x1:Hard	–	–	X	X
12	PTLPS	Poisoned TLP Severity	RW	0:Hard	–	–	X	X
11–6	RESERVED		RO	0	–	–	X	X
5	SDES	Surprise Down Error Severity	RO	0:Hard	–	–	X	X
4	DLPES	Data Link Protocol Error Severity	RW	0x1:Hard	–	–	X	X
3–0	RESERVED		RO	0	–	–	X	X

**PCI EXPRESS CORRECTABLE ERROR STATUS REGISTER (PCICFG_CORR_ERR_STATUS,
OFFSET 0x120)**

Table 92: PCI Express Correctable Error Status Register (pcicfg_corr_err_status, Offset 0x120)

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–14	RESERVED		RO	0	–	–	X	X
13	ANFS	Advisory Non fatal Error Status	WC	0:Hard	–	–	X	X
12	RTTS	Replay Timer Timeout Status	WC	0:Hard	–	–	X	X
11–9	RESERVED		RO	0	–	–	X	X
8	RNRS	REPLAY_NUM Rollover Status	WC	0:Hard	–	–	X	X
7	BDLLPS	Bad DLLP Status	WC	0:Hard	–	–	X	X
6	BTLPS	Bad TLP Status	WC	0:Hard	–	–	X	X
5–1	RESERVED		RO	0	–	–	X	X
0	RES	Receiver Error Status	WC	0:Hard	–	–	X	X

PCI EXPRESS CORRECTABLE ERROR MASK REGISTER (PCICFG_CORR_ERR_MASK, OFFSET 0x124)

Table 93: PCI Express Correctable Error Mask Register (pcicfg_corr_err_mask, Offset 0x124)

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–14	RESERVED		RO	0	–	–	X	X
13	ANFM	Advisory Non fatal Error Mask	RW	0x1:Hard	–	–	X	X
12	RTTS	Replay Timer Timeout Mask	RW	0:Hard	–	–	X	X
11–9	RESERVED		RO	0	–	–	X	X
8	RNRS	REPLAY_NUM Rollover Mask	RW	0:Hard	–	–	X	X
7	BDLLPS	Bad DLLP Mask	RW	0:Hard	–	–	X	X
6	BTLPS	Bad TLP Mask	RW	0:Hard	–	–	X	X
5–1	RESERVED		RO	0	–	–	X	X
0	RES	Receiver Error Mask	RW	0:Hard	–	–	X	X

PCI EXPRESS ADVANCED ERROR CAPABILITY CONTROL REGISTER (PCICFG_ADV_ERR_CAP_CONTROL, OFFSET 0x128)

Table 94: PCI Express Advanced Error Capability Control Register (pcicfg_adv_err_cap_control, Offset 0x128)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–9	RESERVED		RO	0	—	—	X	X
8	ECRCEN	ECRC Check Enable	RW	0:Hard	—	—	X	X
7	ECRCCAP	ECRC Check Capable	RO	0x1	—	—	X	X
6	ECRCGEN	ECRC generate Enable	RW	0:Hard	—	—	X	X
5	ECRCGCap	ECRC generation Capable	RO	0x1	—	—	X	X
4–0	FIRST_UERR_PTR	First Error Pointer—These bits correspond to the bit position in which the first error occurred.	RO	0:Hard	—	—	X	X

PCI EXPRESS HEADER LOG 1 REGISTER (PCICFG_HEADER_LOG1, OFFSET 0x12C)

Description	Mode	Reset	06	08	09	16
The Header Log 1 register returns the first DW of the TLP header associated with an error.	RO	0	—	—	X	X

PCI EXPRESS HEADER LOG 2 REGISTER (PCICFG_HEADER_LOG2, OFFSET 0x130)

Description	Mode	Reset	06	08	09	16
The Header Log 2 register returns the second DW of the TLP header associated with an error.	RO	0	—	—	X	X

PCI EXPRESS HEADER LOG 3 REGISTER (PCICFG_HEADER_LOG3, OFFSET 0x134)

Description	Mode	Reset	06	08	09	16
The Header Log 3 register returns the third DW of the TLP header associated with an error.	RO	0	—	—	X	X

PCI EXPRESS HEADER LOG 4 REGISTER (PCICFG_HEADER_LOG4, OFFSET 0x138)

Description	Mode	Reset	06	08	09	16
The Header Log 4 register returns the fourth DW of the TLP header associated with an error.	RO	0	—	—	X	X

PCI EXPRESS POWER BUDGET CAPABILITY ID REGISTER (PCICFG_PWR_BDGT_CAP_ID, OFFSET 0x150)

Description	Mode	Reset	06	08	09	16
The Power Budget Capability ID indicates the beginning of the Power Budget capability block.	RO	0x4	—	—	X	X

PCI EXPRESS POWER BUDGET CAPABILITIES REGISTERS

PCI EXPRESS POWER BUDGET NEXT CAPABILITY POINTER REGISTER (PCICFG_PWR_BDGT_CAP_OFF, OFFSET 0x152)

Table 95: PCI Express Power Budget Net Capability Pointer Register (pcicfg_pwr_bdgt_cap_off, Offset 0x152)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–4	NEXT	Next Capabilities Pointer.	RO	0x160	—	—	X	X
3–0	VER	Capability ID Version.	RO	0x1	—	—	X	X

PCI EXPRESS POWER BUDGET DATA SELECT REGISTER (PCICFG_PWR_BDGT_DATA_SEL, OFFSET 0x154)

Table 96: PCI Express Power Budget Data Select Register (pcicfg_pwr_bdgt_data_sel, Offset 0x154)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–8	RESERVED		RO	0	—	—	X	X
7–0	DS_VALUE	This value selects the value visible in the Appendix “PCI Express Power Budget Data Register (pcicfg_pwr_bdgt_data, Offset 0x158)”. 	RW	0	—	—	X	X
Value	Name	Description						
0	0	Data Select value 0						
1	1	Data Select value 1						
2	2	Data Select value 2						
3	3	Data Select value 3						
4	4	Data Select value 4						
5	5	Data Select value 5						
6	6	Data Select value 6						
7	7	Data Select value 7						

PCI EXPRESS POWER BUDGET DATA REGISTER (PCICFG_PWR_BDGT_DATA, OFFSET 0x158)*Table 97: PCI Express Power Budget Data Register (pcicfg_pwr_bdgt_data, Offset 0x158)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
31–21	RESERVED		RO	0	–	–	X	X
20–18	RAIL	Power rail	RO	0	–	–	X	X
17–15	TYPE	Type	RO	0	–	–	X	X
14–13	PM_STATE	PM State	RO	0	–	–	X	X
12–10	RESERVED		RO	0	–	–	X	X
9–8	DSCALE	Data Scale	RO	0	–	–	X	X
7–0	BASE_PWR	Base Power	RO	0	–	–	X	X

PCI EXPRESS POWER BUDGET CAPABILITY REGISTER (PCICFG_PWR_BDGT_CAPABILITY, OFFSET 0x15C)*Table 98: PCI Express Power Budget Capability Register (pcicfg_pwr_bdgt_capability, Offset 0x15c)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
31–1	RESERVED		RO	0	–	–	X	X
0	PCIE_CFG_PB_CAP_SYS_ALLOC	When this bit is set it indicates that the power budget for the device is included within the system power budget. Reported Power Budgeting Data for this device should be ignored by software for power budgeting decisions if this bit is set.	RO	0x1	–	–	X	X

PCI EXPRESS VIRTUAL CHANNEL CAPABILITIES REGISTERS

PCI EXPRESS VIRTUAL CHANNEL CAPABILITY ID REGISTER (VC_CAP_ID, OFFSET 0x160)

Description	Mode	Reset	06	08	09	16
The Virtual Channel Capability ID Register indicates the beginning of the Virtual Channel Capability block.	RO	0x2	-	-	X	X

PCI EXPRESS VIRTUAL CHANNEL NEXT CAPABILITY POINTER REGISTER (PCICFG_VC_CAP_OFF, OFFSET 0x162)

Table 99: PCI Express Virtual Channel Next Capability Pointer (pcicfg_vc_cap_off, Offset 0x162)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–4	NEXT	Next Capabilities Pointer	RO	0	-	-	X	X
3–0	VER	Capability ID Version	RO	0x1	-	-	X	X

PCI EXPRESS PORT VIRTUAL CHANNEL CAPABILITIES REGISTERS

PCI EXPRESS PORT VIRTUAL CHANNEL CAPABILITY REGISTER (PCICFG_PORT_VC_CAPABILITY, OFFSET 0x164)

Description	Mode	Reset	06	08	09	16
This register is not implemeted.	RW	0	-	-	X	X

PCI EXPRESS PORT VIRTUAL CHANNEL CAPABILITY 2 REGISTER (PCICFG_PORT_VC_CAPABILITY2, OFFSET 0x168)

Description	Mode	Reset	06	08	09	16
This register is not implemeted.	RW	0	-	-	X	X

PCI EXPRESS PORT VIRTUAL CHANNEL CONTROL REGISTER (PCICFG_PORT_VC_CONTROL, OFFSET 0x16c)

Description	Mode	Reset	06	08	09	16
This register is not implemeted.	RW	0	-	-	X	X

**PCI EXPRESS PORT VIRTUAL CHANNEL STATUS REGISTER (PCICFG_PORT_VC_STATUS,
OFFSET 0x16E)**

Description	Mode	Reset	06	08	09	16
This register is not implemeted.	RW	0	-	-	X	X

**PCI EXPRESS PORT VIRTUAL CHANNEL ARBITRATION TABLE REGISTER
(PCICFG_PORT_ARB_TABLE, OFFSET 0x170)**

Description	Mode	Reset	06	08	09	16
This register is not implemeted.	RW	0	-	-	X	X

**PCI EXPRESS PORT VIRTUAL CHANNEL RESOURCE CONTROL REGISTER
(PCICFG_VC_RSRC_CONTROL, OFFSET 0x174)**

Table 100: PCI Express Port Virtual Channel Resource Control Register (pcicfg_vc_rsrc_control, Offset 0x174)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	VC_ENABLE	Enables virtual channel.	RO	0x1	-	-	X	X
30–8	RESERVED		RO	0	-	-	X	X
7–0	TC_VC_MAP	This field indicates the TCs that are mapped to the VC resource. This field is valid for all devices. Bit 0 of this field is read only. It is set to 1 for the default VC0.	RW	0x1	-	-	X	X

PCI INTERFACE BLOCK REGISTERS

The PCI Interface Block (PCI) provides access to the PCI, PCI-X, or PCI Express block of the NetXtreme II controller.

PCI GRC WINDOW ADDRESS REGISTER (PCI_GRC_WINDOW_ADDR, OFFSET 0x400)*Table 101: PCI GRC Window Address Register (pci_grc_window_addr, Offset 0x400)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	–	–	–
25–8	VALUE	This value controls what portion of the full 4-MB GRC register space is visible in the GRC Window that appears between 32 KB and 64 KB in the PCI address space.	RW	0	X	–	–	–
7–0	RESERVED		RO	0	X	–	–	–
31–21	RESERVED		RO	0	–	X	–	–
20–15	VALUE	This value controls what portion of the full 4-MB GRC register space is visible in the GRC Window that appears between 32 KB and 64 KB in the PCI address space.	RW	0	–	X	–	–
14–0	Reserved		RO	0	–	X	–	–
31	SEP_WIN	Setting this bit splits the GRC window space into four address spaces and automatically splits those address spaces into 8-KB segments.	RO	0	–	–	X	X
30–22	RESERVED		RO	0	–	–	X	X
21–13	VALUE	This value controls what portion of the full 4-MB GRC register space is visible in the GRC Window that appears between 32 KB and 40 KB in the PCI address space.	RW	0	–	–	X	X
12–0	RESERVED		RO	0	–	–	X	X

PCI CONFIGURATION 1 REGISTER (PCI_CONFIG_1, OFFSET 0x404)*Table 102: PCI Configuration 1 Register (pci_config_1, Offset 0x404)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–14	RESERVED		RO	0	X	X	–	–

Table 102: PCI Configuration 1 Register (pci_config_1, Offset 0x404) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
13–11	WRITE_BOUNDARY	Setting this field causes the PCI block to terminate write transactions before crossing specific address boundaries. This type of operation may be used to avoid problems in specific bridges.	RW	0	X	X	–	–
Value Name Description								
	0 OFF	Disable						
	1 16	Stop at 16-byte boundary						
	2 32	Stop at 32-byte boundary						
	3 64	Stop at 64-byte boundary						
	4 128	Stop at 128-byte boundary						
	5 256	Stop at 256-byte boundary						
	6 512	Stop at 512-byte boundary						
	7 1024	Stop at 1024-byte boundary						
10–8	READ_BOUNDARY	Setting this field causes the PCI block to terminate read transactions before crossing specific address boundaries. This type of operation may be used to avoid problems in specific bridges.	RW	0	X	X	–	–
Value Name Description								
	0 OFF	Disable						
	1 16	Stop at 16-byte boundary						
	2 32	Stop at 32-byte boundary						
	3 64	Stop at 64-byte boundary						
	4 128	Stop at 128-byte boundary						
	5 256	Stop at 256-byte boundary						
	6 512	Stop at 512-byte boundary						
	7 1024	Stop at 1024-byte boundary						
7–0	RESERVED		RO	0	X	X	–	–
31–0	RESERVED		RO	0	–	–	X	X

PCI CONFIGURATION 2 REGISTER (PCI_CONFIG_2, OFFSET 0x408)**Table 103: PCI Configuration 2 Register (pci_config_2, Offset 0x408)**

Bit	Name	Description	Mode	Reset	06	08	09	16														
31–26	RESERVED		RO	0	X	X	X	X														
25	KEEP_REQ_ASSERT	When this bit is 1, the PCI block will keep the <u>REQ</u> signal asserted until the DMA Engine block completes the transaction. When this bit is 0, the PCI block will deassert request every time it is granted the bus and will have to re-acquire the bus after each disconnect. This bit must be set to 0 in some emulation environments.	RW	0x1	X	X	—	—														
24	FORCE_32_BIT_TGT	Setting this bit forces the interface to operate in 32-bit mode and does not force the high half of the BUS to drive as if the chip was booted into 32-bit mode. This bit only effects the response to target cycles for the mailbox queue, since all other target cycles are 32-bits only.	RW	0	X	X	—	—														
23	FORCE_32_BIT_MSTR	Setting this bit forces the interface to operate in 32-bit mode and does not force the high half of the BUS to drive as if the chip was booted into 32-bit mode. This bit only effects the generation of REQ64 on bus master cycles.	RW	0	X	X	—	—														
22–21	MAX_READ_LIMIT	The minimum value of either this field or the MAX_MEM_READ field of the “PCI-X Command Register (pcicfg_pcix_command, Offset 0x42)” on page 162 is used to limit the maximum size of Memory Read commands attempted by the PCI-X interface. The encoding of these two fields are the same.	RW	0	X	X	—	—														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>512</td> <td>Limit reads to 512-bytes max read size.</td> </tr> <tr> <td>1</td> <td>1K</td> <td>Limit reads to 1024-bytes max read size.</td> </tr> <tr> <td>2</td> <td>2K</td> <td>Limit reads to 2048-bytes max read size.</td> </tr> <tr> <td>3</td> <td>4K</td> <td>Limit reads to 4096-bytes max read size.</td> </tr> </tbody> </table>	Value	Name	Description	0	512	Limit reads to 512-bytes max read size.	1	1K	Limit reads to 1024-bytes max read size.	2	2K	Limit reads to 2048-bytes max read size.	3	4K	Limit reads to 4096-bytes max read size.					
Value	Name	Description																				
0	512	Limit reads to 512-bytes max read size.																				
1	1K	Limit reads to 1024-bytes max read size.																				
2	2K	Limit reads to 2048-bytes max read size.																				
3	4K	Limit reads to 4096-bytes max read size.																				
20–16	MAX_SPLIT_LIMIT	The minimum value of either this field or the MAX_SPLIT field of the “PCI-X Command Register (pcicfg_pcix_command, Offset 0x42)” on page 162 is used to limit the number of outstanding DMA read requests that will be attempted. Unlike the MAX_SPLIT value, this value is not encoded so that any value between 0 and 31 may be used.	RW	0	X	X	—	—														
25–17	RESERVED		RO	0	—	—	X	X														
16	BAR_PREFETCH	Setting this bit will set the PREFETCH bit of the “Base Address 1 Register (pcicfg_bar_1, Offset 0x10)” on page 156.	RO	0	—	—	X	X														

Table 103: PCI Configuration 2 Register (pci_config_2, Offset 0x408) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–8	EXP_ROM_SIZE	These bits control the size of the Expansion ROM area advertised in the “ Expansion ROM Base Address Register (pcicfg_exp_rom_bar, Offset 0x30) ” on page 159. When this value is non-0, the Expansion ROM attention must be handled by an internal processor to move data between the NVRAM and the Expansion ROM interface. This value is sticky and only reset by a hard reset.	RW	0	X	X	X	X
		Value Name Description						
		0 DISABLED –						
		1 1K 1 KB						
		2 2K 2 KB						
		3 4K 4 KB						
		4 8K 8 KB						
		5 16K 16 KB						
		6 32K 32 KB						
		7 64K 64 KB						
		8 128K 128 KB						
		9 256K 256 KB						
		10 512K 512 KB						
		11 1M 1 MB						
		12 2M 2 MB						
		13 4M 4 MB						
		14 8M 8 MB						
		15 16M 16 MB						
7	FIRST_CFG_DONE	After a PCI reset occurs, this bit is set when the first PCI configuration cycle access is performed by the PCI block. This may be used by firmware to detect if the host already has the reset values of the configuration space. This may happen if the NVM system is much slower than expected. In this case, the firmware can choose to not exist or show an error on LEDs, etc. instead of changing configuration space values that the host has already read. Unlike almost all the other bits in this register area, this bit is reset by PCI Reset.	RO	0	X	X	X	X
6	CFG_CYCLE_RETRY	This bit forces the PCI bus to re-try all cycles to the configuration space until it is cleared. This is used to block the host from accessing context if needed to prevent reading of false data. This bit may be used in combination with the FIRST_CFG_DONE bit to prevent changing of the configuration space values after they have been read by the system. Typically, this bit is set by the firmware while the configuration space is programmed.	RW	0	X	X	X	X

Table 103: PCI Configuration 2 Register (pci_config_2, Offset 0x408) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16																																																		
5	EXP_ROM_RETRY	This bit forces the PCI bus to retry all cycles to the current Expansion ROM BAR area. When this bit is set, then no Expansion ROM attention is generated. This bit must be cleared to allow the attention to be generated.	RW	0	X	X	X	X																																																		
4	BAR1_64ENA	When this bit is set, it enables the advertisement of “ Base Address 1 Register (pcicfg_bar_1, Offset 0x10) ” on page 156 as a 64-bit address. The value of this bit maps directly to bit 2 of the Base Address 1 register. This bit is sticky and only reset by a hard reset.	RW	0x1	X	X	X	X																																																		
3–0	BAR1_SIZE	These bits control the size of the ADDRESS area advertised in the “ Base Address 1 Register (pcicfg_bar_1, Offset 0x10) ” on page 156. This value is sticky and only reset by a hard reset.	RW	0x3	X	X	X	X																																																		
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr><td>0</td><td>DISABLED</td><td>This value causes the base address area to be completely read-only so that no memory space is indicated.</td></tr> <tr><td>1</td><td>64K</td><td>64 KB</td></tr> <tr><td>2</td><td>128K</td><td>128 KB</td></tr> <tr><td>3</td><td>256K</td><td>256 KB</td></tr> <tr><td>4</td><td>512K</td><td>512 KB</td></tr> <tr><td>5</td><td>1M</td><td>1 MB</td></tr> <tr><td>6</td><td>2M</td><td>2 MB</td></tr> <tr><td>7</td><td>4M</td><td>4 MB</td></tr> <tr><td>8</td><td>8M</td><td>8 MB</td></tr> <tr><td>9</td><td>16M</td><td>16 MB</td></tr> <tr><td>10</td><td>32M</td><td>32 MB</td></tr> <tr><td>11</td><td>64M</td><td>64 MB</td></tr> <tr><td>12</td><td>128M</td><td>128 MB</td></tr> <tr><td>13</td><td>256M</td><td>256 MB</td></tr> <tr><td>14</td><td>512M</td><td>512 MB</td></tr> <tr><td>15</td><td>1G</td><td>1 GB</td></tr> </tbody> </table>	Value	Name	Description	0	DISABLED	This value causes the base address area to be completely read-only so that no memory space is indicated.	1	64K	64 KB	2	128K	128 KB	3	256K	256 KB	4	512K	512 KB	5	1M	1 MB	6	2M	2 MB	7	4M	4 MB	8	8M	8 MB	9	16M	16 MB	10	32M	32 MB	11	64M	64 MB	12	128M	128 MB	13	256M	256 MB	14	512M	512 MB	15	1G	1 GB					
Value	Name	Description																																																								
0	DISABLED	This value causes the base address area to be completely read-only so that no memory space is indicated.																																																								
1	64K	64 KB																																																								
2	128K	128 KB																																																								
3	256K	256 KB																																																								
4	512K	512 KB																																																								
5	1M	1 MB																																																								
6	2M	2 MB																																																								
7	4M	4 MB																																																								
8	8M	8 MB																																																								
9	16M	16 MB																																																								
10	32M	32 MB																																																								
11	64M	64 MB																																																								
12	128M	128 MB																																																								
13	256M	256 MB																																																								
14	512M	512 MB																																																								
15	1G	1 GB																																																								

PCI CONFIGURATION 3 REGISTER (PCI_CONFIG_3, OFFSET 0x40c)*Table 104: PCI Configuration 3 Register (pci_config_3, Offset 0x40c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	PCI_POWER	This bit indicates the current state of power on the PCI bus. If this bit is: <ul style="list-style-type: none">• 1, it indicates that the PCI pad ring has power• 0, it indicates that the PCI pad ring does not have power (D3 Cold)	RO	0	X	X	X	X
30	VAUX_PRESET	This bit indicates the input level on the VAUX_PRESET pin. This indicates if the VAUX supply is available in the current configuration. The value also controls the value of the PME_IN_D3_COLD field of the "Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)" on page 166 .	RO	X	X	X	X	X
29	RESERVED	–	RO	0	X	X	X	X
28–27	PM_STATE	This field returns the last value written to the STATE field of the Power Management Control/Status Register (see "Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)" on page 167).	RO	0	X	X	X	X
26	PME_ENABLE	This field returns the current state of the PME_ENABLE bit of the "Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)" on page 167 . This bit is a sticky bit and only reset by a hard reset.	RO	0	X	X	X	X
25	PME_STATUS	This field returns the current value of the PME_STATUS bit of the "Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)" on page 167 . This bit is a sticky bit and only reset by a hard reset.	RO	0	X	X	X	X
24	FORCE_PME	Setting this bit to 1 forces the PME pin to be asserted regardless of any programming in the MAC. This simulates the PME event. The PME control bits in the configuration space still control the output normally. This bit is sticky and only reset by a hard reset.	RW	0	X	X	X	X
23–8	RESERVED		RO	0	X	X	–	–
23–16	RESERVED		RO	0	–	–	X	X
15–8	REG_STICKY_BYTE	This value is only reset by REG_HARD_RST.	RW	0	–	–	X	X
7–0	STICKY_BYTE	This value is only reset by hard reset such that it can be used to detect initial power up if a non-0 value is written by the firmware after initialization. It has no hardware function other than reset type detection.	RW	0:Hard	X	X	X	X

PCI POWER MANAGEMENT DATA A REGISTER (PCI_PM_DATA_A, OFFSET 0x410)

This register controls the first 4 power management PM_Data read values

Table 105: PCI Power Management Data A Register (pci_pm_data_a, Offset 0x410)

Bit	Name	Description	Mode	Reset
31–24	PM_DATA_3_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 3. This is the power consumed in D3 state. This value is sticky and only reset by a hard reset.	RW	0
23–16	PM_DATA_2_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 2. This is the power consumed in D2 state. This value is sticky and only reset by a hard reset.	RW	0
15–8	PM_DATA_1_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 1. This is the power consumed in D1 state. This value is sticky and only reset by a hard reset.	RW	0
7–0	PM_DATA_0_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 0. This is the power consumed in D0 state. This value is sticky and only reset by a hard reset.	RW	0

PCI POWER MANAGEMENT DATA B REGISTER (PCI_PM_DATA_B, OFFSET 0x414)

This register controls the second four power management PM_Data read values.

Table 106: PCI Power Management Data B Register (pci_pm_data_b, Offset 0x414)

Bit	Name	Description	Mode	Reset
31–24	PM_DATA_7_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167X is set to 7. This is the power dissipated in D3 state. This value is sticky and only reset by a hard reset.	RW	0
23–16	PM_DATA_6_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 6. This is the power dissipated in D2 state. This value is sticky and only reset by a hard reset.	RW	0
15–8	PM_DATA_5_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 5. This is the power dissipated in D1 state. This value is sticky and only reset by a hard reset.	RW	0
7–0	PM_DATA_4_PRG	This field controls the value read from the “Power Management Data Register (pcicfg_pm_data, Offset 0x4f)” on page 168 when the DATA_SEL field of the “Power Management Control/Status Register (pcicfg_pm_csr, Offset 0x4c)” on page 167 is set to 4. This is the power dissipated in D0 state. This value is sticky and only reset by a hard reset.	RW	0

PCI SWAP DIAGNOSTICS 0 REGISTER (PCI_SWAP_DIAG0, OFFSET 0x418)

Description	Mode	Reset
This register is used to help diagnose the swapping characteristics of the system. By reading this RO register and adjusting the TARGET_BYTE_SWAP and TARGET_WORD_SWAP fields in the “Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68)” on page 172, the driver can set the target access to map word and byte accesses correctly for target accesses. This can then be used to diagnose the DMA swapping by writing known patterns into internal NetXtreme II memory and performing DMA operations on them.	0x010203 04	

PCI SWAP DIAGNOSTICS 1 REGISTER (PCI_SWAP_DIAG1, OFFSET 0x41C)

Description	Mode	Reset
This register is used in conjunction with the “PCI Swap Diagnostics 0 Register (pci_swap_diag0, RO Offset 0x418)” on page 209 to determine the swapping characteristics of the system.	0x111213 14	

PCI EXPANSION ROM ADDRESS REGISTER (PCI_EXP_ROM_ADDR, OFFSET 0x420)*Table 107: PCI Expansion ROM Address Register (pci_exp_rom_addr, Offset 0x420)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	REQ	This bit will be set if there is a pending request for action by the firmware to handle a PCI BAR memory read request.	RO	0	X	X	-	-
30–24	RESERVED		RO	0	X	X	-	-
23–2	ADDRESS	This value is the address requested by the current (or last) PCI ROM Expansion cycle request. When the PCI block detects a decode of the ROM BAR area, it will place the offset from the BAR value in this register and retry the PCI bus to make the requester wait.	RO	0	X	X	-	-
1–0	RESERVED		RO	0	X	X	-	-
31–0	RESERVED		RO	0	-	-	X	X

PCI EXPANSION ROM DATA REGISTER (PCI_EXP_ROM_DATA, OFFSET 0x424)

Description	Mode	Reset	06	08	09	16
When this register is written, the PCI block will present the data written to the PCI WO bus and clear the REQ bit in the “PCI Expansion ROM Address Register (pci_exp_rom_addr, Offset 0x420)” on page 210.	0		X	X	-	-

PCI VPD INTERFACE REGISTER (PCI_VPD_intf, OFFSET 0x428)*Table 108: PCI VPD Interface Register (pci_vpd_intf, Offset 0x428)*

Bit	Name	Description	Mode	Reset
31–1	RESERVED		RO	0
0	INTF_REQ	This bit will be set if there is a pending request for action by the firmware to handle a Vital Product Data interface.	RO	0

PCI VPD ADDRESS FLAG REGISTER (PCI_VPD_ADDR_FLAG, OFFSET 0x42E)*Table 109: PCI VPD Address Flag Register (pci_vpd_addr_flag, 0x42e)*

Bit	Name	Description	Mode	Reset
15	WR	This bit indicates if the host is requesting a read or a write cycle. If this bit is set, then the host has requested the data in the “PCI VPD Data Register (pcicfg_vpd_data, Offset 0x430)” on page 211 to be passed to the NVM interface. If the value is clear, then the host has requested the data to be passed from the NVM interface to the PCI VPD Data register. The value of this bit is only valid if the INTF_REQ bit is set.	RW	X
14–2	ADDRESS	This value is the byte address of the VPD value being requested by the host in the FLAG field of the “VPD Address/Flag Register (pcicfg_vpd_flag_addr, Offset 0x52)” on page 169. Since the data register is 32-bits wide, the bottom 2 bits of this register should always be written as zeros.	RW	X
1–0	RESERVED		RO	0

PCI VPD DATA REGISTER (PCI_VPD_DATA, OFFSET 0x430)

Description	Mode	Reset
This register is the data register for passing values between the NVM interface and the “VPD Data Register (pcicfg_vpd_data, Offset 0x54)” on page 169. When the WR bit of the “PCI VPD Address Flag Register (pci_vpd_addr_flag, 0x42e)” on page 211 is clear, this word should be written with the NVM data requested in the ADDRESS value to clear the INTF_REQ bit of the “PCI VPD Interface Register (pcicfg_vpd_intf, Offset 0x428)” on page 210. When the WR bit is set, this word should be read and written to the NVM interface. After the NVM interface write is complete, this value should be written with the same value to clear the INTF_REQ bit.	RW	0xXXXXXX XXX

PCI ID VALUE 1 REGISTER (PCI_ID_VAL1, OFFSET 0x434)*Table 110: PCI ID Value 1 Register (pci_id_val1, Offset 0x434)*

Bit	Name	Description	Mode	Reset	06	08	09	16
15–0	DEVICE_ID	This register programs the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159. (BCM5706C)	RW	0x164a:Hard	X	–	–	–
15–0	DEVICE_ID	This register programs the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159. (BCM5706S)	RW	0x16aa:Hard	X	–	–	–
15–0	DEVICE_ID	This register programs the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159. (BCM5708C)	RW	0x164c:Hard	–	X	–	–

Table 110: PCI ID Value 1 Register (pci_id_val1, Offset 0x434) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–0	DEVICE_ID	This register programs the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159. (BCM5708S)	RW	0x16ac:Hard	X	—	—	—
15–0	DEVICE_ID	This register programs the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159. (BCM5709C, BCM5716)	RW	0x1639:Hard	—	—	X	X
15–0	DEVICE_ID	This register programs the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159. (BCM5709S)	RW	0x163a:Hard	—	—	X	—

PCI ID VALUE 2 REGISTER (PCI_ID_VAL2, OFFSET 0x438)**Table 111: PCI ID Value 2 Register (pci_id_val2, Offset 0x438)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	SUBSYSTEM_ID	This value controls the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159 in the configuration space.	RW	0x164a:Hard	X	—	—	—
31–16	SUBSYSTEM_ID	This value controls the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159 in the configuration space.	RW	0x164a:Hard	—	X	—	—
31–16	SUBSYSTEM_ID	This value controls the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159 in the configuration space.	RW	0x164a:Hard	—	—	X	X
31–16	SUBSYSTEM_ID	This value controls the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159 in the configuration space.	RW	0x164a:Hard	—	—	X	X
15–0	SUBSYSTEM_VENDOR_ID	This value controls the read value of the “Subsystem ID Register (pcicfg_subsystem_id, Offset 0x2e)” on page 159 in the configuration space.	RW	0x14e4:Hard				

PCI ID VALUE 3 REGISTER (PCI_ID_VAL3, OFFSET 0x43C)*Table 112: PCI ID Value 3 Register (pci_id_val3, Offset 0x43c)*

Bit	Name	Description	Mode	Reset
31–24	REVISION_ID	This register programs the read value of the “Revision ID Register (pcicfg_revision_id, Offset 0x08)” on page 155. The hardware default value is 0.	RW	0:Hard
23–0	CLASS_CODE	This register programs the read value of the “Class Code Register (pcicfg_class_code, Offset 0x09)” on page 155. The 24-bit Class Code register identifies the generic function of the device. All of the legal values are specified in the PCI specification. The default value for this register is the class code for an Ethernet interface (0x020000).	RW	0x20000:Hard

PCI ID VALUE 4 REGISTER (PCI_ID_VAL4, OFFSET 0x440)*Table 113: PCI ID Value 4 Register (pci_id_val4, Offset 0x440)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–29	RESERVED		RO	0	X	X	–	–
31–16	RESERVED		RW	0:Hard	–	–	X	X
28–26	MAX_CUMULATIVE_SIZE	This value controls the read value of the MAX_CUM_SIZE field of the “PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)” on page 163. The default is 3, which indicates 8 KB of data.	RW	0x3:Hard	X	–	–	–
25–23	MAX_SPLIT_SIZE	This value controls the read value of the MAX_SPLIT field of the “PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)” on page 163. The default is 4, which indicates 8 split transactions.	RW	0x4:Hard	X	–	–	–
22–21	MAX_MEM_READ_SIZE	This value controls the read value of the MAX_MEM_READ field of the “PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)” on page 163. The default is 0, which indicates a 512-byte read size.	RW	0:Hard	X	–	–	–
20–18	RESERVED		RO	0	X	X	–	–
17	MAX_133_ADVVERTISE	This value controls the read value of the 133_MHZ field of the “PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)” on page 163. The default is 1, which indicates 133-MHz capability.	RW	0x1:Hard	X	X	–	–
16	MAX_64_ADVVERTISE	This value controls the read value of the 64_BIT field of the “PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)” on page 163. The default is 1, which indicates 64-bit capability.	RW	0x1:Hard	X	X	–	–
15	MSI_ENABLE	This bit indicates the programming of the ENABLE field “MSI Control Register (pcicfg_msi_control, Offset 0x5a)” on page 170. If this bit is set, it means that the interrupt output is masked and all interrupts must be indicated with MSI cycles.	RO	0	X	X	X	X
14–12	MSI_ADVVERTISE	This value controls the read value of the MCAP field of the “MSI Control Register (pcicfg_msi_control, Offset 0x5a)” on page 170. The default is 0, which is one MSI.	RW	0:Hard	X	X	–	–
14–12	MULTI_MSG_CAP	This value controls the read value of the MSI CTRL MCAP value in the PCI configuration space. The default is 0, which is one MSI. This value is sticky and only reset by HARD Reset.	RW	0x3	–	–	X	X
11–9	MSI_LIMIT	This value reports the MENA field of the “MSI Control Register (pcicfg_msi_control, Offset 0x5a)” on page 170. This value will always be equal or less than what was advertised.	RO	0:Hard	X	X	X	X

Table 113: PCI ID Value 4 Register (*pci_id_val4*, Offset 0x440) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
8	RESERVED		RO	0	X	X	—	—
8	MSI_PV_MASK_CAP	This value controls the per vector masking capability in the MSI control field. This value is sticky and only reset by a hard reset.	RW	0:Hard	—	—	X	X
7–6	PM_SCALE_PRG	This value is read as the DATA_SCALE field of the “Power Management Control/Status Register (<i>pcicfg_pm_csr</i> , Offset 0x4c)” on page 167.	RW	1:Hard	X	X	X	X
Value Name Description								
	0	Indicate unknown power consumption.						
	1	Indicate values specified in .1W increments.						
	2	Indicate values specified in .01W increments.						
	3	Indicate values specified in .001W increments.						
5–4	RESERVED		RO	0	X	X	X	X
3–0	CAP_ENA	This value controls the read value of the next capability pointers in the PCI configuration space and allows each extra capability to be independently disabled by manipulation of the next pointer values. The read values for each enable combination are shown as follows. Bit 0 enables the PCIX capability. Bit 1 enables the Power Management capability. Bit 2 enables the VPD capability, and Bit 3 enables the MSI capability.	RW	0xf:Hard	X	X	—	—
Value Name Description								
	0	None						
	1	PCI-X						
	2	PM						
	3	PM, PCI-X						
	4	VPD						
	5	VPD, PCI-X						
	6	VPD, PM						
	7	VPD, PM, PCI-X						
	8	MSI						
	9	MSI, PCI-X						
	10	MSI, PM						

Table 113: PCI ID Value 4 Register (pci_id_val4, Offset 0x440) (Cont.)

Bit	Name	Description				Mode	Reset	06	08	09	16
3–0 (cont.)	CAP_ENA	11	11	MSI, PM, PCI-X		RW	0xf:Hard	X	X	–	–
		12	12	MSI, VPD							
		13	13	MSI, VPD, PCI-X							
		14	14	MSI, VPD, PM							
		15	15	MSI, VPD, PM, PCI-X							
3–0	CAP_ENA	This value controls the read value of the next capability pointers in the PCIe configuration space and allows each extra capability to be independently disabled by manipulation of the next pointer values. The read values for each enable combination is shown as follows. PCIe capability is always enabled. Bit 0 enables the Power Management capability. Bit 1 enables the VPD capability, and Bit 2 enables the MSI capability and Bit 3 is MSIX capability. This value is sticky and only reset by hard reset.				RW	0xf:Hard	–	–	X	X
Value Name Description											
0	0	PCIe									
1	1	MSIX, PCIe									
2	2	MSI, PCIe									
3	3	MSI, MSIX, PCIe									
4	4	VPD, PCIe									
5	5	VPD, MSIX, PCIe									
6	6	VPD, MSI, PCIe									
7	7	VPD, MSI, MSIX, PCIe									
8	8	PM, PCIe									
9	9	PM, MSIX, PCIe									
10	10	PM, MSI, PCIe									
11	11	PM, MSI, MSIX, PCIe									
12	12	PM, VPD, PCIe									
13	13	PM, VPD, MSIX, PCIe									
14	14	PM, VPD, MSI, PCIe									
15	15	PM, VPD, MSI, MSIX, PCIe									

PCI ID VALUE 5 REGISTER (PCI_ID_VAL5, OFFSET 0x444)*Table 114: PCI ID Value 5 Register (pci_id_val5, Offset 0x444)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–6	RESERVED		RO	0	X	X	—	—
31–10	RESERVED		RW	0	—	—	X	X
9	NO_SOFT_RESET_PM_CS R	This indicates function does not perform an internal reset when RW 0x1: Hard transitioning from D3 to D0. The value is reflected in the corresponding field in PM capabilities register.	RO	0	—	—	X	X
8–6	PM_VERSION	The value indicates the function complies with which revision of RW 0x3: Hard PCI PM specification. This value is reflected in the corresponding field in PM capabilities register.	RO	0	—	—	X	X
5	PME_IN_D3_HOT	This bit indicates whether the device supports asserting PME from the D3hot power state. It is reflected in the PME_IN_D3_HOT bit of the “Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)” on page 166.	RW	0x1:Hard	X	X	X	X
4	PME_IN_D2	This bit indicates whether the device supports asserting PME from the D2 power state. It is reflected in the PME_IN_D2 bit of the “Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)” on page 166.	RW	0:Hard	X	X	X	X
3	PME_IN_D1	This bit indicates whether the device supports asserting PME from the D1 power state. It is reflected in the PME_IN_D1 bit of the “Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)” on page 166.	RW	0:Hard	X	X	X	X
2	PME_IN_D0	This bit indicates whether the device supports asserting PME from the D0 power state. It is reflected in the PME_IN_D0 bit of the “Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)” on page 166.	RW	0:Hard	X	X	X	X
2	PME_IN_D0	This bit indicates whether the device supports asserting PME from the D0 power state. It is reflected in the PME_IN_D0 bit of the “Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)” on page 166.	RW	0x1:Hard	—	—	X	X

Table 114: PCI ID Value 5 Register (pci_id_val5, Offset 0x444) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	D2_SUPPORT	This bit indicates whether the device supports the D2 power management state. It is reflected in the D2_SUPPORT bit of the "Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)" on page 166.	RW	0:Hard	X	X	X	X
0	D1_SUPPORT	This bit indicates whether the device supports the D1 power management state. It is reflected in the D1_SUPPORT bit of the "Power Management Capabilities Register (pcicfg_pm_capability, Offset 0x4a)" on page 166.	RW	0:Hard	X	X	X	X

PCI PCI-X EXTENDED STATUS REGISTER (PCI_PCIX_EXTENDED_STATUS, OFFSET 0x448)**Table 115: PCI PCI-X Extended Status Register (pci_pcix_extended_status, Offset 0x448)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	SPLIT_COMP_MSG_IDX	This value indicates the PCI-X Split Completion Message Index that was received on the last PCIX Split completion. This value is valid when the SPLIT_ERR bit of the "PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)" on page 163 is set.	RO	0	X	X	—	—
23–20	RESERVED		RO	0	X	X	—	—
19–16	SPLIT_COMP_MSG_CLAS S	This value indicates the PCI-X Split Completion Message Class that was received on the last PCI-X Split completion. This value is valid when the SPLIT_ERR bit of the "PCI-X Status Register (pcicfg_pcix_status, Offset 0x44)" on page 163 is set.	RO	0	X	X	—	—
15–10	RESERVED		RO	0	X	X	—	—
9	LONG_BURST	This bit enables the long burst mode for the PCI block. A similar bit in the DMA Engine must also be set for the mode to work. This mode will automatically chain DMA operations together that were broken into multiple pieces for DMA Engine buffering reasons.	RW	0	X	X	—	—
8	NO_SNOOP	This bit controls the value of the No Snoop attribute in PCI-X transactions. Normally, this bit should be cleared.	RW	0	X	X	—	—
7–0	RESERVED		RO	0	X	X	—	—

PCI ID VALUE 6 REGISTER (PCI_ID_VAL6, OFFSET 0x44c)*Table 116: PCI ID Value 6 Register (pci_id_val6, Offset 0x44c)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–24	RESERVED		RO	0	X	X	X	X
23–16	BIST	This register controls the read value of the “BIST Register (pcicfg_bist, Offset 0x0f)” on page 156.	RW	0:Hard	X	X	X	X
15–8	MIN_GNT	This register controls the read value of the “Minimum Grant Register (pcicfg_min_grant, Offset 0x3e)” on page 161.	RW	0x40:H ard	X	X	–	–
7–0	MAX_LAT	This register controls the read value of the “Maximum Latency Register (pcicfg_maximum_latency, Offset 0x3f)” on page 161.	RW	0:Hard	X	X	–	–
15–0	RESERVED		RO	0	–	–	X	X

PCI MSI DATA REGISTER (PCI_MSI_DATA, OFFSET 0x450)*Table 117: PCI MSI Data Register (pci_msi_data, Offset 0x450)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–16	RESERVED		RO	0	X	X	X	X
15–0	MSI_DATA	This register reflects the “MSI Data Register (pcicfg_msi_data, Offset 0x64)” on page 171. The completion processor may use this value to determine the address it will use for vectored MSI cycles.	RO	0	X	X	X	X

PCI MSI ADDRESS HIGH REGISTER (PCI_MSI_ADDR_H, OFFSET 0x454)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
This register reflects the value of the “MSI Address High Register (pcicfg_msi_addr_h, Offset 0x60)” on page 171. The completion processor (COM) may use this value to determine the address it will use for vectored MSI cycles.	RO	0	X	X	X	X

PCI MSI ADDRESS LO REGISTER (PCI_MSI_ADDR_L, OFFSET 0x458)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
This register reflects the value of the “MSI Address Low Register (pcicfg_msi_addr_l, Offset 0x5c)” on page 171. The completion processor (COM) may use this value to determine the address it will use for vectored MSI cycles.	RO	0	X	X	X	X

PCI CONFIGURATION ACCESS COMMAND REGISTER (PCI_CFG_ACCESS_CMD, OFFSET 0x45C)

Table 118: PCI Configuration Access Command Register (pci_cfg_access_cmd, Offset 0x45c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	–	–	X	X
31–28	WR_REQ	When any of these bits are written as 1, an indirect write of the configuration space is started at the address specified in ADDR field, using the data in the “PCI Configuration Access Data Register (pci_cfg_access_data, Offset 0x460)” on page 365. The value of WR_REQ acts as the byte-enable settings for that access. When the cycle is complete, these bits return to 0. All the WR_REQ and RD_REQ bits must be read as 0 before any of the WR_REQ bits are written as 1. Note: Only those items that are documented as writable in the PCI configuration space definition may be written through this interface.	SC	0	X	X	–	–
27	RD_REQ	When this bit is written as 1, a 32-bit indirect read of the configuration space is initiated at the address specified in the ADDR field. When the cycle is complete, the data is available in the “PCI Configuration Access Data Register (pci_cfg_access_data, Offset 0x460)” on page 365 and this bit returns to 0. All of the WR_REQ and RD_REQ bits must be read as 0 before the RD_REQ bit is written as 1.	SC	0	X	X	–	–
26–8	RESERVED		RO	0	X	X	–	–
7–2	ADR	This field specifies the 32-bit word address of the Configuration Space register to be accessed when the WR_REQ or RD_REQ fields are written as 1. This value can be written on the same cycle that the corresponding WR_REQ and RD_REQ fields are written as 1. “Register Window Data Register (pcicfg_reg_window, Offset 0x80)” on page 176 and the “Mailbox Queue Data Register (pcicfg_mailbox_queue_data, Offset 0x94)” on page 178 cannot be accessed through this method. Writes have no effect and reads of these registers return zeros.	RW	0	X	X	–	–
1–0	RESERVED		RO	0	X	X	–	–

MISCELLANEOUS (MISC) BLOCK REGISTERS

MISC COMMAND REGISTER (MISC_COMMAND, OFFSET 0x800)

Table 119: MISC Command Register (misc_command, Offset 0x800)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–23	RESERVED		RO	0	X	X	—	—
31–30	RESERVED		RO	0	—	—	X	X
29	POWERUP_EVENT	This bit indicates that a power-up event has occurred (i.e. reset_force_perst_b has transitioned from 0 to 1).	WC	0:CMN	—	—	X	X
28	PCIE_DIS	This bit reports the current state of the PCIE DIS pin. If this bit is '1', it means that the LOM design has been strapped to support management only. The PCI power will always read as '0' in this state, as if the chip is in Out-Of-Box WOL mode. Based on this bit the firmware may or may not enable magic packet based on customer requirements. [SHARE]	RO	0:CMN	—	—	X	X
27	PCIE_LINK_IN_L23	This bit indicates that the link is prepared for power removal. [SHARE]	RO	0:CMN	—	—	X	X
26	DINTEG_ATTN_EN	When set to 1, attention events based on parity or data integrity errors will be enabled.	RW	0:TC	—	—	X	X
25	SHUTDOWN_EN	When set to 1, shutdown based on parity or data integrity errors will be enabled. This enable bit has no control over the SW_SHUTDOWN field.	RW	0:TC	—	—	X	X
24	SW_SHUTDOWN	Writing this bit as a '1' will shutdown the respective toe core. This is not masked by the SHUTDOWN EN bit.	RW	0:TC	—	—	X	X
23	POWERDOWN_EVENT	This bit indicates that a powerdown event has occurred (i.e., reset_force_perst_b has transitioned from 1 to 0). Write 1 to clear.	WC	0:CMN	—	—	X	X
22–16	PAR_ERR_RAM	If the PAR_ERROR bit is high, then this register indicates the RAM that first caused the parity error condition. The numeric value corresponds to the bit position of the RAM in the “ MISC PERR Enable 0 Register (misc_perr_ena0, Offset 0x8a4) ” on page 258, “ MISC PERR Enable 1 Register (misc_perr_ena1, Offset 0x8a8) ” on page 260, or “ MISC PERR Enable 1 Register (misc_perr_ena1, Offset 0x8a8) ” on page 260.	RO	0:TC	X	X	X	X
15–12	RESERVED				X	X	—	—

Table 119: MISC Command Register (misc_command, Offset 0x800) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–12	CS16_ERR_LOC	If the CS16 ERR bit is set this register indicates the checker that first caused the cs16 error condition. The numeric value corresponds to the bit position of the checker in the cs16_ena register. This value will stay latched until all the error status is cleared from the cs16_ena register.	RO	0:TC	–	–	X	X
11–10	RESERVED				X	X	X	X
9	RESERVED		RO	0	X	X	–	–
9	CS16_ERR	When this bit is set it indicates that an enabled cs16 checker has detected an error. The number of the checker to generate the error is indicated in CS16_ERR_LOC field. This bit is the generator for the cs16 attention generation. When the Error condition is entered, all I/O of the chip is suspended except for status blocks to the host and target PCI accesses. This bit is cleared by clearing all of the error status bits in the “MISC Checksum 16 Error Register (misc_cs16_err, offset 0x8e0)” on page 270.	RO	0:TC	–	–	X	X
8	PAR_ERROR	This bit is set when an enabled RAM has generated a parity error. The number of the first RAM to generate the parity error is indicated in PAR_ERR_RAM field. This bit is the value for the Parity Error Attention condition. When the Parity Error condition is entered, all I/O of the chip is suspended except for status blocks to the host and target PCI writes. Setting this bit will clear the parity error condition and allow blocks to process normally again.	WC	0	0	0	0	0
7–6	RESERVED		RO	0	X	X	–	–
7	CMN_SW_RESET	Reading this bit after writing a'1' to it causes an internal reset of the common core. The reset is not actually issued in the chip until the read of the bit is done in order to ensure that there are no pending posted transactions to the chip while the reset is being performed. Software should wait 500 ns after issuing the read before performing any other accesses to the device. This reset should not be used by firmware unless it is certain there are no ongoing PCIE transactions. This reset is essentially for debug purposes and should be used carefully. [SHARE]	SC	0:CMN	–	–	X	X
6	HD_RESET	Setting this bit will cause the chip to do a hard reset like the HARD_RESET bit except that the sticky bits in the PCI function are not reset. [SHARE]	SC	0:CMN	–	–	X	X

Table 119: MISC Command Register (*misc_command*, Offset 0x800) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
5	HARD_RESET	Setting this bit will cause the chip to do an internal reset exactly like a power-on reset. There is no protection for this request and it may cause any current PCI cycles to lock up. This reset is intended for use under manufacturing conditions only. [SHARE]	SC	0:CMN	X	X	X	X
4	CORE_RESET	Setting this bit will cause the device to perform an internal core reset. No PCI cycles should be executed for at least 100 PCI clocks after this bit is set. This bit should not be used by firmware. This bit is used for debugging purposes only and should not be used in production code. See the CORE_RST_REQ bit of the Miscellaneous Configuration Register (see “ Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68) ” on page 302) for the correct method to initialize the device from a host driver.	SC	0:	X	X	—	—
4	SW_RESET	Reading this bit after setting it causes an internal reset of the corresponding TOE core. The reset is not actually issued in the chip until the read of the bit is done to ensure that there are no pending posted transactions to the chip while the reset is being performed. Software should wait 500 ns after issuing the read before performing any other accesses to the device. This reset should not be used by firmware unless it is certain there are no ongoing PCIe transactions.	SC	0	—	—	X	X
3-2	RESERVED		RO	0	X	X	X	X
1	DISABLE_ALL	Setting this bit will disable all the blocks in the chip. This bit is used for debugging purposes only and should not be used in production code. Refer to the CORE_RST_REQ bit of the Miscellaneous Configuration Register (see “ Miscellaneous Configuration Register (pcicfg_misc_config, Offset 0x68) ” on page 172) for the correct method to initialize the device from a host driver.	SC	0	X	X	X	X
0	ENABLE_ALL	Setting this bit will enable all the blocks in the chip for normal operation. Set this bit after all the blocks are configured.	SC	0	X	X	X	X

MISC CONFIGURATION REGISTER (MISC_CFG, OFFSET 0x804)

Table 120: MISC Configuration Register (misc_cfg, Offset 0x804)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–11	RESERVED		RO	0	X	–	–	–
31–13	RESERVED		RO	0	–	X	–	–
31–15	RESERVED		RO	0	–	–	X	X
14	PARITY MODE	Setting this bit changes the parity checking on the RAMs from even parity to odd parity. [SHARE]	RW	0:CM N	–	–	X	X
13	PORT SELECT	Setting this bit selects between PORT0 and PORT1. All the SELECT registers are selected by this bit. [SHARE]	RW	0:TC	–	–	X	X
12	DBU_GRC_TMO_UT	Setting this bit enables a 1-ms timer in the GRC block to timeout any access from the DBU (UART) block that does not finish within 512 ms. When this bit is cleared, this timeout is disabled. If this timeout occurs, the DBU (UART) will be acked with a read data value of 0xdeadbee1. No attention will be asserted on this timeout.	RW	0	–	X	–	–
12	RESERVED		RO	0	–	–	X	X
11	MCP_GRC_TMO_UT	Setting this bit enables a 1-ms timer in the GRC block to timeout any access from the MCP block that does not finish within 512 ms. When this bit is cleared, this timeout is disabled. If this timeout occurs, the MCP will be acked with a read data value of 0xdeadbee2. No attention will be asserted on this timeout.	RW	0	–	X	–	–
11–8	LEDMODE	These bits control the LED mux setting at the top level. This mux chooses LED source.	RW	0x1:T C	–	–	X	X
Value	Name	Description						
0	MAC	LED outputs of the MAC block are connected to LED pins. Open drain emulation, active low, no quality indication.						
1	PHY1	LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open drain emulation, active low, no quality indication.						
2	PHY2	LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open drain emulation, active low, with quality indication for the GPHY port. Identical to PHY1 mode for the SerDes. Open drain emulation, active low.						
3	PHY3	Tri-Color mode. Two bit encoded LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open drain emulation, active low, no quality indication.						
4	MAC2	LED outputs of the MAC block are connected to LED pins. Open source emulation, active high, no quality indication.						

Table 120: MISC Configuration Register (*misc_cfg*, Offset 0x804) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
11–8 (cont.)	LEDMODE	5	PHY4	LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open source emulation, active high, no quality indication.	RW	0x1:T C	–	–	X	X
		6	PHY5	LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open source emulation, active high, with quality indication for the GPHY port. Identical to PHY4 mode for the SerDes.						
		7	PHY6	Tri-Color mode. Two bit encoded LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open source emulation, active high, no quality indication.						
		8	MAC3	Speed indication LED outputs of the MAC block are logically OR-ed with inverted activity signal (as it is active low). Open drain emulation, active low, no quality indication.						
		9	PHY7	LED outputs of the GPHY or SerDes blocks are logically OR-ed with the inverted activity signal (as it is active low). Open drain emulation, active low, no quality indication.						

Table 120: MISC Configuration Register (*misc_cfg*, Offset 0x804) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
11–8 (cont.)	LEDMODE	10	PHY8	Tri-Color mode. Two bit encoded LED outputs of the GPHY or SerDes blocks are connected to LED pins. Open drain emulation, active low, logically OR-ed link and activity, no quality indication.	RW	0x1:T C	—	—	X	X
		11	PHY9	Bi-Color mode for GPHY. Two bit encoded LED outputs of the GPHY is connected to LED pins. Direct drive, no quality indication.						
		12	MAC4	Speed indication LED outputs of the MAC block are logically OR-ed with the inverted activity signal (as it is active low) and inverted resulting signal connected to the corresponding LED pin. Open source emulation, active high, no quality indication.						
		13	PHY10	LED outputs of the GPHY or SerDes blocks are logically OR-ed with the inverted activity signal (as it is active low) and inverted resulting signal is connected to the corresponding LED pin. Open source emulation, active high, no quality indication.						
		14	PHY11	Tri-Color mode for SerDES. Two bit encoded LED outputs of the SerDes block are connected to LED pins. Open source emulation, active high, logically OR-ed link and activity, no quality indication.						
		15	UNUSE D	Unused. Defaults to PHY1 mode.						

Table 120: MISC Configuration Register (*misc_cfg*, Offset 0x804) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
10–8	LEDMODE	These bits control the LED mux setting at the top level. This mux chooses the LED source.	RW	0	X	X	–	–
		Value	Name	Description				
		0	MAC	When this value is set, the EMAC LED controls is selected (see “EMAC LED Register (<i>emac_led</i> , Offset 0x140c)” on page 317).				
		1	GPHY1	When this value is set, the GPHY LED outputs has the following definitions for the Copper PHY:				
				<ul style="list-style-type: none"> • SPD1000 is lit when a 1-Gb link is active. • SPD100 is lit when a 100-Mbit link is active. • LINKLED is lit when a 10-MB link is active. • TRAFFICLED is lit when transmitting or receiving data. <p>And from the SerDes as:</p> <ul style="list-style-type: none"> • SPD1000 = light when 2.5-GB link. • SPD100 = light when 1G or 100-MB link. • LINKLED = light when 100-MB or 10-MB link. • TRAFFIC = light when transmit or receive data. • All outputs are open drain drive. 				
		2	GPHY2	When this value is set, the GPHY LED outputs has the following meanings:				
				<ul style="list-style-type: none"> • SPD1000 is lit when a 1-GB link is active. • SPD100 is lit when a 100-MB link is active and not quality LED. • LINKLED is lit when a link is active and not quality. • TRAFFIC is lit when transmitting or receiving data. <p>For the SerDes mode, the outputs are encoded the same as for GPHY1 mode. All LED outputs will be active low, totem pole</p> <ul style="list-style-type: none"> • All outputs are active low, totem pole drive. 				
7–5	RESERVED		RO	0	–	–	X	X
7	CLK_CTRL_OVERRIDE	When this bit is set, the Clock Control register (see “MISC Clock Control Register (<i>misc_clock_control_bits</i> , 0x818)” on page 236) overrides the value provided by the PCI clock domain.	RW	0	X	X	–	–

Table 120: MISC Configuration Register (misc_cfg, Offset 0x804) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16															
6	BYPASS_EJTAG	This bit configures the JTAG pins to control the on-board Boundary SCAN JTAG controller only. The on-board processor EJTAG SCAN JTAG controller is bypassed.	RW	0	X	-	-	-															
6	RESERVED		RO	0	-	X	X	X															
5	BYPASS_BSCAN	This bit configures the JTAG pins to control the on-board processor JTAG controllers only. The Boundary SCAN JTAG controller is bypassed.	RW	0	X	-	-	-															
5	RESERVED		RO	0	-	X	X	X															
4	CK25_OUT_ALT_SRC	When this bit is set the CK25_OUT pin drives clocks other than CK25 reference output. [SHARE]	RW	0:CM N	X	X	X	X															
3	BIST_EN	This bit globally enables the BIST logic. When set to 1, all RAM BIST logic is connected to RAMs and register files in lieu of the normal functional logic. This bit must be 0 for normal operation. [SHARE]	RW	0	X	X	X	X															
2–1	NVM_WR_EN	These bits control NVRAM write access. [SHARE]	RW	0	X	X	X	X															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td><td>PROTECT</td><td>When this value is set, the NVM is protected from all writes.</td></tr> <tr> <td>1</td><td>PCI</td><td>When this value is set, the NVM only allows writes when PCI_RST# is high.</td></tr> <tr> <td>2</td><td>ALLOW</td><td>When this value is set, the NVM allows writes using normal NVM commands.</td></tr> <tr> <td>3</td><td>ALLOW</td><td>When this value is set, the NVM allows writes using normal NVM commands.</td></tr> </tbody> </table>	Value	Name	Description	0	PROTECT	When this value is set, the NVM is protected from all writes.	1	PCI	When this value is set, the NVM only allows writes when PCI_RST# is high.	2	ALLOW	When this value is set, the NVM allows writes using normal NVM commands.	3	ALLOW	When this value is set, the NVM allows writes using normal NVM commands.						
Value	Name	Description																					
0	PROTECT	When this value is set, the NVM is protected from all writes.																					
1	PCI	When this value is set, the NVM only allows writes when PCI_RST# is high.																					
2	ALLOW	When this value is set, the NVM allows writes using normal NVM commands.																					
3	ALLOW	When this value is set, the NVM allows writes using normal NVM commands.																					
0	PCI_GRC_TMOU_T	Setting this bit enables a 1-ms timer in the GRC block to timeout any access from the PCI block that does not finish within about 300 ms. When this bit is cleared, this timeout is disabled. If this timeout occurs, the GRC will assert its attention output. This attention can only be cleared by a CORE reset. [OR]	RW	0:TC	X	X	X	X															

MISC ID REGISTER (MISC_ID, OFFSET 0x808)*Table 121: MISC ID Register (misc_id, Offset 0x808)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	CHIP_NUM	These bits indicate the part number for the chip.	RO	0x5706	X	—	—	—
				0x5708	—	X	—	—
				0x5709	—	—	X	X
15–12	CHIP_REV	These bits indicate the base revision of the chip.	RO	0xX	X	X	X	X
11–4	CHIP_METAL	These bits indicate the metal revision of the chip.	RO	0xXX	X	X	X	X
3–0	BOND_ID	These bits indicate the Pad ring bond BOND_ID. This value can be modified at build time to allow different packages to be identified from software. The pads for the BOND_ID must be implemented as pull-up pads and bond wires installed to the GND to make them 0 inputs for the initial version of each chip. This will ensure that bonding is possible for each BOND_ID pin. These bits are connected to the OTP ROM MEDIARO CONTROL[3:0] outputs. BOND ID[3:0] indicate type of the device.	RO	0xX	X	X	—	—
3–0	BOND_ID	Value	Name	Description	RO	0	—	— X X
		0	X	All 4 PHYs are available.				
		3	C	Two Copper PHYs are available.				
		12	S	Two SerDes PHYs are available.				

MISC ENABLE STATUS REGISTER (MISC_ENABLE_STATUS_BITS, OFFSET 0x80c)

This register indicates the current enable status (without LFSR modulation) for each block. A 1 indicates that the block is enabled and a 0 indicates that the block is disabled.

Table 122: MISC Enable Status Register (misc_enable_status_bits, Offset 0x80c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–29	RESERVED		RO	0	X	X	X	X
28	RESERVED		RO	0	X	X	–	–
28	RV2P_CMD_SCHEDULER_ENABLE	This bit controls or indicates status for the Command Scheduler block independently from all other blocks.	RO	0	–	–	X	X
27	UMP_ENABLE	This bit indicates the status for the UMP sub-block of the Management processor.	RO	0	X	X	–	–
27	RESERVED		RO	0	–	–	X	X
26	DMA_ENGINE_ENABLE	This bit indicates the status for the DMA Engine block independently from all other blocks. [OR]	RO	0	X	X	X	X
25	TIMER_ENABLE	This bit indicates the status for the Timer block independently from all other blocks.	RO	0	X	X	X	X
24	MGMT_PROCESSOR_ENABLE	This bit indicates the status for the Management Processor block independently from all other blocks. [OR]	RO	0	X	X	X	X
23	CMD_PROCESSOR_ENABLE	This bit indicates the status for the Command Processor block independently from all other blocks.	RO	0	X	X	X	X
22	CMD_SCHEDULER_ENABLE	This bit indicates the status for the Command Scheduler block independently from all other blocks.	RO	0	X	X	X	X
21	CONTEXT_ENABLE	This bit indicates the status for the Context block independently from all other blocks.	RO	0	X	X	X	X
20	MAILBOX_QUEUE_ENABLE	This bit indicates the status for the Mailbox Queue block independently from all other blocks.	RO	0	X	X	X	X
19	HOST_COALESCE_ENABLE	This bit indicates the status for the Host Coalesce block independently from all other blocks.	RO	0	X	X	X	X
18	COMPLETION_ENABLE	This bit indicates the status for the Completion block independently from all other blocks.	RO	0	X	X	X	X
17	RX_DMA_ENABLE	This bit indicates the status for the RX DMA block independently from all other blocks.	RO	0	X	X	X	X
16	RX_BD_CACHE_ENABLE	This bit indicates the status for the RX BD Cache block independently from all other blocks.	RO	0	X	X	X	X

Table 122: MISC Enable Status Register (misc_enable_status_bits, Offset 0x80c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15	RX_V2P_ENABLE	This bit indicates the status for the RX V2P block independently from all other blocks.	RO	0	X	X	X	X
14	RX_PROCESSOR_ENABLE	This bit indicates the status for the RX Processor block independently from all other blocks.	RO	0	X	X	X	X
13	RX_LOOKUP_ENABLE	This bit indicates the status for the RX Lookup block independently from all other blocks.	RO	0	X	X	X	X
12	RX_MBUF_ENABLE	This bit indicates the status for the RX MBUF block independently from all other blocks.	RO	0	X	X	X	X
11	RX_PARSER_CATCHUP_ENABLE	This bit indicates the status for the RX Parser Catchup block independently from all other blocks.	RO	0	X	X	X	X
10	RX_PARSER_MAC_ENABLE	This bit indicates the status for the RX Parser MAC block independently from all other blocks.	RO	0	X	X	X	X
9	EMAC_ENABLE	This bit indicates the status for the EMAC block independently from all other blocks.	RO	0	X	X	X	X
8	TX_ASSEMBLER_ENABLE	This bit indicates the status for the TX Assembler block independently from all other blocks.	RO	0	X	X	X	X
7	TX_HEADER_Q_ENABLE	This bit indicates the status for the TX Header Queue sub-block of the TX Assembler independently from all other blocks.	RO	0	X	X	X	X
6	TX_PAYLOAD_Q_ENABLE	This bit indicates the status for the TX Payload Queue sub-block of the TX Assembler independently from all other blocks.	RO	0	X	X	X	X
5	TX_PATCHUP_ENABLE	This bit indicates the status for the TX Patchup block independently from all other blocks.	RO	0	X	X	X	X
4	TX_DMA_ENABLE	This bit indicates the status for the TX DMA block independently from all other blocks.	RO	0	X	X	X	X
3	TX_PROCESSOR_ENABLE	This bit indicates the status for the TX Processor block independently from all other blocks.	RO	0	X	X	X	X
2	TX_BD_CACHE_ENABLE	This bit indicates the status for the TX BD Cache block independently from all other blocks.	RO	0	X	X	X	X
1	TX_BD_READ_ENABLE	This bit indicates the status for the TX BD Read block independently from all other blocks.	RO	0	X	X	X	X
0	TX_SCHEDULER_ENABLE	This bit indicates the status for the TX Scheduler block independently from all other blocks.	RO	0	X	X	X	X

MISC ENABLE SET REGISTER (MISC_ENABLE_SET_BITS, OFFSET 0x810)

This register allows individual blocks to be enabled. Each bit that is written as a 1 will enable the corresponding block. These bits always read as 0.

Table 123: MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-29	RESERVED		RO	0	X	X	X	X
28	RESERVED		RO	0	X	X	-	-
28	RV2P_CMD_SCHEDULER_ENABLE	This bit controls or indicates status for the Command Scheduler block independently from all other blocks.	SC	0	-	-	X	X
27	RESERVED		RO	0	-	-	X	X
27	UMP_ENABLE	This bit controls the UMP sub-block of the Management processor.	SC	0	X	X	-	-
26	DMA_ENGINE_ENABLE	This bit controls the DMA Engine block independently from all other blocks. [OR]	SC	0	X	X	X	X
25	TIMER_ENABLE	This bit controls the Timer block independently from all other blocks.	SC	0	X	X	X	X
24	MGMT_PROCESSOR_ENABLE	This bit controls the Management Processor block independently from all other blocks. [OR]	SC	0	X	X	X	X
23	CMD_PROCESSOR_ENABLE	This bit controls the Command Processor block independently from all other blocks.	SC	0	X	X	X	X
22	CMD_SCHEDULER_ENABLE	This bit controls the Command Scheduler block independently from all other blocks.	SC	0	X	X	X	X
21	CONTEXT_ENABLE	This bit controls the Context block independently from all other blocks.	SC	0	X	X	X	X
20	MAILBOX_QUEUE_ENABLE	This bit controls the Mailbox Queue block independently from all other blocks.	SC	0	X	X	X	X
19	HOST_COALESCE_ENABLE	This bit controls the Host Coalesce block independently from all other blocks.	SC	0	X	X	X	X
18	COMPLETION_ENABLE	This bit controls the Completion block independently from all other blocks.	SC	0	X	X	X	X
17	RX_DMA_ENABLE	This bit controls the RX DMA block independently from all other blocks.	SC	0	X	X	X	X
16	RX_BD_CACHE_ENABLE	This bit controls the RX BD Cache block independently from all other blocks.	SC	0	X	X	X	X
15	RX_V2P_ENABLE	This bit controls the RX V2P block independently from all other blocks.	SC	0	X	X	X	X
14	RX_PROCESSOR_ENABLE	This bit controls the RX Processor block independently from all other blocks.	SC	0	X	X	X	X
13	RX_LOOKUP_ENABLE	This bit controls the RX Lookup block independently from all other blocks.	SC	0	X	X	X	X

Table 123: MISC Enable Set Register (misc_enable_set_bits, Offset 0x810) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	RX_MBUF_ENABLE	This bit controls or indicates status for the RX MBUF block independently from all other blocks.	SC	0	X	X	X	X
11	RX_PARSER_CATCHUP_ENA_BLE	This bit controls the RX Parser Catchup block independently from all other blocks.	SC	0	X	X	X	X
10	RX_PARSER_MAC_ENABLE	This bit controls the RX Parser MAC block independently from all other blocks.	SC	0	X	X	X	X
9	EMAC_ENABLE	This bit controls the EMAC block independently from all other blocks.	SC	0	X	X	X	X
8	TX_ASSEMBLER_ENABLE	This bit controls the TX Assembler block independently from all other blocks.	SC	0	X	X	X	X
7	TX_HEADER_Q_ENABLE	This bit controls the TX Header Queue sub-block of the TX Assembler independently from all other blocks.	SC	0	X	X	X	X
6	TX_PAYLOAD_Q_ENABLE	This bit controls the TX Payload Queue sub-block of the TX Assembler independently from all other blocks.	SC	0	X	X	X	X
5	TX_PATCHUP_ENABLE	This bit controls the TX Patchup block independently from all other blocks.	SC	0	X	X	X	X
4	TX_DMA_ENABLE	This bit controls the TX DMA block independently from all other blocks.	SC	0	X	X	X	X
3	TX_PROCESSOR_ENABLE	This bit controls the TX Processor block independently from all other blocks.	SC	0	X	X	X	X
2	TX_BD_CACHE_ENABLE	This bit controls the TX BD Cache block independently from all other blocks.	SC	0	X	X	X	X
1	TX_BD_READ_ENABLE	This bit controls the TX BD Read block independently from all other blocks.	SC	0	X	X	X	X
0	TX_SCHEDULER_ENABLE	This bit controls the TX Scheduler block independently from all other blocks.	SC	0	X	X	X	X

MISC ENABLE CLEAR REGISTER (MISC_ENABLE_CLR_BITS, OFFSET 0x814)

This register allows individual blocks to be disabled. Each bit that is written as a 1 will disable the corresponding block. These bits always read as 0.

Table 124: MISC Enable Clear Register (misc_enable_clr_bits, 0x814)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–29	RESERVED		RO	0	X	X	X	X
28	RESERVED		RO	0	X	X	—	—
28	RV2P_CMD_SCHEDULER_ENABLE	This bit controls or indicates status for the Command Scheduler block independently from all other blocks.	SC	0	—	—	X	X
27	RESERVED		RO	0	—	—	X	X
27	UMP_ENABLE	This bit controls the UMP sub-block of the Management processor.	SC	0	X	X	X	X
26	DMA_ENGINE_ENABLE	This bit controls the DMA Engine block independently from all other blocks. [OR]	SC	0	X	X	X	X
25	TIMER_ENABLE	This bit controls the Timer block independently from all other blocks.	SC	0	X	X	X	X
24	MGMT_PROCESSOR_ENABLE	This bit controls the Management Processor block independently from all other blocks. [OR]	SC	0	X	X	X	X
23	CMD_PROCESSOR_ENABLE	This bit controls the Command Processor block independently from all other blocks.	SC	0	X	X	X	X
22	CMD_SCHEDULER_ENABLE	This bit controls the Command Scheduler block independently from all other blocks.	SC	0	X	X	X	X
21	CONTEXT_ENABLE	This bit controls the Context block independently from all other blocks.	SC	0	X	X	X	X
20	MAILBOX_QUEUE_ENABLE	This bit controls the Mailbox Queue block independently from all other blocks.	SC	0	X	X	X	X
19	HOST_COALESCE_ENABLE	This bit controls the Host Coalesce block independently from all other blocks.	SC	0	X	X	X	X
18	COMPLETION_ENABLE	This bit controls the Completion block independently from all other blocks.	SC	0	X	X	X	X
17	RX_DMA_ENABLE	This bit controls the RX DMA block independently from all other blocks.	SC	0	X	X	X	X
16	RX_BD_CACHE_ENABLE	This bit controls the RX BD Cache block independently from all other blocks.	SC	0	X	X	X	X
15	RX_V2P_ENABLE	This bit controls the RX V2P block independently from all other blocks.	SC	0	X	X	X	X
14	RX_PROCESSOR_ENABLE	This bit controls the RX Processor block independently from all other blocks.	SC	0	X	X	X	X

Table 124: MISC Enable Clear Register (misc_enable_clr_bits, 0x814) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
13	RX_LOOKUP_ENABLE	This bit controls the RX Lookup block independently from all other blocks.	SC	0	X	X	SP	SP
12	RX_MBUF_ENABLE	This bit controls the RX MBUF block independently from all other blocks.	SC	0	X	X	SP	SP
11	RX_PARSER_CATCHUP_ENABLE	This bit controls the RX Parser Catchup block independently from all other blocks.	SC	0	X	X	SP	SP
10	RX_PARSER_MAC_ENABLE	This bit controls the RX Parser MAC block independently from all other blocks.	SC	0	X	X	SP	SP
9	EMAC_ENABLE	This bit controls the EMAC block independently from all other blocks.	SC	0	X	X	SP	SP
8	TX_ASSEMBLER_ENABLE	This bit controls the TX Assembler block independently from all other blocks.	SC	0	X	X	SP	SP
7	TX_HEADER_Q_ENABLE	This bit controls the TX Header Queue sub-block of the TX Assembler independently from all other blocks.	SC	0	X	X	SP	SP
6	TX_PAYLOAD_Q_ENABLE	This bit controls the TX Payload Queue sub-block of the TX Assembler independently from all other blocks.	SC	0	X	X	SP	SP
5	TX_PATCHUP_ENABLE	This bit controls the TX Patchup block independently from all other blocks.	SC	0	X	X	SP	SP
4	TX_DMA_ENABLE	This bit controls the TX DMA block independently from all other blocks.	SC	0	X	X	SP	SP
3	TX_PROCESSOR_ENABLE	This bit controls the TX Processor block independently from all other blocks.	SC	0	X	X	SP	SP
2	TX_BD_CACHE_ENABLE	This bit controls the TX BD Cache block independently from all other blocks.	SC	0	X	X	SP	SP
1	TX_BD_READ_ENABLE	This bit controls the TX BD Read block independently from all other blocks.	SC	0	X	X	SP	SP
0	TX_SCHEDULER_ENABLE	This bit controls the TX Scheduler block independently from all other blocks.	SC	0	X	X	SP	SP

MISC CLOCK CONTROL REGISTER (MISC_CLOCK_CONTROL_BITS, OFFSET 0x818)

This register is the CORE clock domain control register. There is a separate PCI domain control register in the “[PCI Clock Control Register \(pcicfg_pci_clock_control_bits, Offset 0x70\)](#)” on page 174. The values of these two registers are logically ORed together to create the actual clock control register value. The read-only bits are valid in both registers. This register is reset by a CORE reset.

Table 125: MISC Clock Control Register (misc_clock_control_bits, 0x818)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–20	RESERVED		RW	0	X	X	–	–
31–28	CORE_CLK_PLL_SPEED	These bits select bits 3:0 of the feed-forward divider of the CORE CLK PLL. The upper bits of the divider are hardwired to 0. The CORE CLK PLL output frequency is equal to VCO/([divider setting]+2) where the VCO frequency is determined by CORE CLK PLL VCO. [SHARE] <i>Example:</i> If the bits in this register are set to 0b0100 then the divider setting is 0b000100 (4), so the PLL output frequency is VCO/6 which is 200 MHz for the VCO running at 1.2 GHz. The divider setting is controllable from 2–17. These bits should only be changed when an alternate clock is selected (CORE CLK ALT =’1’) and software should only switch back to the PLL source 350 µs after changing.	RW	0x4	–	–	X	X
27	RESERVED		RW	0	X	X	X	X
26–24	CORE_CLK_PLL_VCO	These bits select bits 2:0 of the feedback divider of the CORE CLK PLL. The upper bits are hard-wired to 0b000010. The VCO frequency is equal to 25 MHz*2*([divider setting]+4). For example, if these bits are set to 0b100, the divider setting is 0b10100 (20), so the VCO runs at 1.2 GHz. The VCO range is thus controllable from 1 GHz–1.35 GHz in 50-MHz increments. These bits should only be changed when an alternate clock is selected (CORE CLK ALT =’1’) and software should only switch back to the PLL source 350 µs after changing these bits. These bits have affect when main power is up or down. [SHARE]	RW	0x4	–	–	X	X
23–18	RESERVED		RW	0	–	–	X	X
17	CORE_CLK_ALT_MGMT	This bit selects the CORE CLK source when main power is not available (PERST B low). When set, it is used to select the 25-MHz clock in place of the PLL output to reduce power. This bit only has affect when main power is down. The firmware should set this bit based on whether or not the PLL is to be used when in mgmt mode. CORE CLK ALT should be used instead of this bit when switching the PLL speed. [SHARE]	RW	0x1	–	–	X	X

Table 125: MISC Clock Control Register (misc_clock_control_bits, 0x818) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
19	USE_SPD_DET	When this bit is set, it changes the PCI PLL strapping to be controlled by the speed detect logic rather than the PCI-X clock speed pin strappings as defined in the PCI-X specification.	RW	0	X	-	-	-
19	RESERVED		RO	0	-	X	X	X
18	RESERVED		RW	0	X	X	X	X
17	PCI_PLL_STOP	This bit powers down the PCI PLL when it is set. This is only used for hardware debugging.	RW	0	X	-	-	-
17	RESERVED		RO	0	-	X	X	X
16	CORE_CLK_PLL_STOP	This bit stops the CORE_CLK PLL when it is set. One of the alternate clocks must be selected when this is done. (ALT = 1) [SHARE]	RW	0	X	X	X	X
15–12	CORE_CLK_PLL_SPEED	These bits control the core clock PLL settings to select different core clock speeds. The clock generator will pick the slowest of the speeds enabled. If no bits are set, then the PLL will run at 100-MHz core clock speed. These bits must only be modified when the ALT bit is 1. The firmware/software must guarantee that the ALT bit changes and the PLL_SPEED changes are not in the same PCI or GRC cycle. The Hardware must only modify the PLL output divisor with this value and not the PLL feedback programming. At reset, this value is to 0x2 if the PCI Pad power detect indicates no PCI pad power, otherwise, this bit resets to 0x0.	RW	0	X	X	-	-
Value Name Description								
0	100	100 MHz (PLL Divisor == 4)						
1	80	80 MHz (PLL Divisor == 5)						
2	50	50 MHz (PLL Divisor == 8)						
4	40	40 MHz (PLL Divisor == 10)						
8	25	25 MHz (PLL Divisor == 16)						
15–12	RESERVED		RW	0	-	-	X	X
11	PLAY_DEAD	This bit disables all CORE_CLK and CPU_CLK for blocks and is used to place the chip into play-dead mode used to minimize VAUX power usage when not configured for VAUX PME assertion. Setting this bit in the MISC block is not a good idea as the PCI cycle may not finish. This bit also sets IDDQ mode for all devices in the chip other than the CK25 oscillator, 1.2V regulator, PME pad and GPIO2 pad.	RW	0	X	X	X	X

Table 125: MISC Clock Control Register (misc_clock_control_bits, 0x818) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16														
10–8	CORE_CLK_ALT_SRC	These bits select the alternate clock source. The clock generator will pick the slowest of the speeds enabled. These bits must only be modified when the ALT bit is 0.	RW	0	X	X	–	–														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>UNDEF</td> <td>Undefined alternate clock select.</td> </tr> <tr> <td>1</td> <td>12</td> <td>Select 25-MHz Reference Clock to make a 12.5-MHz core clock.</td> </tr> <tr> <td>2</td> <td>6</td> <td>Select 12.5-MHz Clock to make a 6.25-MHz core clock.</td> </tr> <tr> <td>4</td> <td>62</td> <td>Select 125-MHz PHY Clock to make 62.5-MHz core clock.</td> </tr> </tbody> </table>	Value	Name	Description	0	UNDEF	Undefined alternate clock select.	1	12	Select 25-MHz Reference Clock to make a 12.5-MHz core clock.	2	6	Select 12.5-MHz Clock to make a 6.25-MHz core clock.	4	62	Select 125-MHz PHY Clock to make 62.5-MHz core clock.					
Value	Name	Description																				
0	UNDEF	Undefined alternate clock select.																				
1	12	Select 25-MHz Reference Clock to make a 12.5-MHz core clock.																				
2	6	Select 12.5-MHz Clock to make a 6.25-MHz core clock.																				
4	62	Select 125-MHz PHY Clock to make 62.5-MHz core clock.																				
10–8	RESERVED		RW	0	–	–	X	X														
7	CORE_CLK_ALT	This bit selects the alternate CORE_CLK source. This is used to select a non-PLL source while the PLL is being manipulated or to select a non-PLL source during WOL mode so the PLL can be powered down all together. [SHARE]	RW	0	X	X	X	X														
6	CORE_CLK_DISABLE	This bit disables CORE_CLK and CPU_CLK to all blocks that are not needed for WOL operation. At reset, this bit is set to 1 if the VAUX_PRESNT bit input is 1 and the PCI Pad power detect indicates no PCI pad power for the PCI version of this register only. Otherwise, this bit resets to 0. The MISC block version of this register always resets this bit to 0.	RW	X	X	X	X	X														
5–4	RESERVED		RO	0	X	X	X	X														

Table 125: MISC Clock Control Register (misc_clock_control_bits, 0x818) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
3–0	PCI_CLK_SPD_DET	This value indicates the detected speed of the PCI clock as compared to the 25-MHz crystal clock.			RO	0x7	X	X	–	–
Value Name Description										
0	32 MHz	PCI Clock is in the range of 28.1 MHz to 39.1 MHz.								
1	38 MHz	PCI Clock is in the range of 34.3 MHz to 45.3 MHz.								
2	48 MHz	PCI Clock is in the range of 40.6 MHz to 41.6 MHz.								
3	55 MHz	PCI Clock is in the range of 46.8 MHz to 60.9 MHz.								
4	66 MHz	PCI Clock is in the range of 56.2 MHz to 71.9 MHz.								
5	80 MHz	PCI Clock is in the range of 67.1 MHz to 90.5 MHz.								
6	95 MHz	PCI Clock is in the range of 85.9 MHz to 103 MHz.								
7	133MHz	PCI Clock is MHz the range of 98.4 MHz to 198 MHz.								
0xf	LOW	PCI Clock is in the range of 0 MHz to 28.2 MHz.								
3–0	RESERVED				RW	0	–	–	X	X

MISC_SPIO REGISTER (MISC_SPIO, OFFSET 0x81c)

This register controls and reads status of the Shared Programmable I/O (SPIO) pins.

Some of the SPIO pins have the following functions:

- SPIO 0—VAUX Enable, when pulsed low, enables supply from VAUX
- SPIO 1—VAUX Disable, when pulsed low, disables supply from VAUX. This is done to disconnect the chip from VAUX power when it will not be generating PME in an adapter application.
- SPIO 2—Control to power switching logic. Drive low to select VAUX supply regardless of VMAIN power availability.
- SPIO 6—Bit 0 of UMP device ID select, read by UMP firmware
- SPIO 7—Bit 1 of UMP device ID select, read by UMP firmware

For the BCM5709 and BCM5716, all bits are of the type [SHARE] except for SPIO2 which is [SPLIT] is asserted when the port is ready for power down. Otherwise, the control of these pins is identical to BCM5706 and BCM5708.

Table 126: MISC_SPIO Register (misc_spio, Offset 0x81c)

Bit	Name	Description	Mode	Reset
31–24	FLOAT	When any of these bits is written as a 1, the corresponding SPIO bit will turn off its drivers and become an input. This is the reset state of all SPIO pins. The read value of these bits will be a 1 if that last command (SET, CLR, or FLOAT) for this bit was a FLOAT.	RW	0xff
23–16	CLR	When any of these bits is written as a 1, the corresponding SPIO bit will drive low. The read value of these bits will be a 1 if that last command (SET, CLR, or FLOAT) for this bit was a CLR.	RW	0
15–8	SET	When any of these bits is written as a 1, the corresponding SPIO bit will drive high (if it has that capability). The read value of these bits will be a 1 if that last command (SET, CLR, or FLOAT) for this bit was a SET.	RW	0
7–0	VALUE	These bits indicate the read value from each of the eight SPIO pins. This is the result value of the pin, not the drive value. Writing these bits will have no effect.	RO	X

MISC SPIO INTERRUPT REGISTER (MISC_SPIO_INT, OFFSET 0x820)

The first four SPIO bits are capable of edge detection for generation of interrupts in any processor. Each input can be handled independently by any of the processors.



Note: For the BCM5709 and BCM5716 controllers, SPIO bits [7:4] are capable of edge detection for generation of interrupts in any processor. Each input can be handled independently by any of the processors.

Table 127: MISC SPIO Interrupt Register (misc_spio_int, Offset 0x820)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	—	—
31–24	OLD_CLR	Writing a 1 to these bits clears the corresponding bit in the OLD_VALUE field. This will acknowledge an interrupt on the falling edge of the corresponding SPIO input. [SHARE]	SC	0	—	—	X	X
27–24	OLD_CLR	Writing a 1 to these bits clears the corresponding bit in the OLD_VALUE field. This will acknowledge an interrupt on the falling edge of the corresponding SPIO input.	SC	0	X	X	—	—
23–20	RESERVED		RO	0	X	X	—	—
23–16	OLD_SET	Writing a 1 to these bits sets the corresponding bit in the OLD_VALUE field. This will acknowledge an interrupt on the rising edge of the corresponding SPIO input. [SHARE]	SC	0	—	—	X	X
19–16	OLD_SET	Writing a 1 to these bits sets the corresponding bit in the OLD_VALUE field. This will acknowledge an interrupt on the rising edge of the corresponding SPIO input.	SC	0	X	X	—	—
15–12	RESERVED		RO	0	X	X	—	—
15–8	OLD_VALUE	These bits indicate the old value of the SPIO input value. When the INT_STATE field is set, this bit indicates the OLD value of the pin such that if INT_STATE is set and this bit is 0, then the interrupt is due to a low to high edge. If INT_STATE is set and this bit is 1, then the interrupt is due to a high to low edge. [SHARE]	RO	X	—	—	X	X
11–8	OLD_VALUE	These bits indicate the old value of the SPIO input value. When the INT_STATE field is set, this bit indicates the OLD value of the pin such that if INT_STATE is set and this bit is 0, then the interrupt is due to a low to high edge. If INT_STATE is set and this bit is 1, then the interrupt is due to a high to low edge	RO	X	X	X	—	—
7–4	RESERVED		RO	0	X	X	—	—
7–0	INT_STATE	These bits indicate the current SPIO interrupt state for each SPIO pin. This bit is cleared when the appropriate OLD_SET or OLD_CLR field bit is written. This bit is set when the SPIO input does not match the current value in OLD_VALUE. [SHARE]	RO	X	—	—	X	X
3–0	INT_STATE	These bits indicate the current SPIO interrupt state for each SPIO pin. This bit is cleared when the appropriate OLD_SET or OLD_CLR field bit is written. This bit is set when the SPIO input does not match the current value in OLD_VALUE.	RO	X	X	X	—	—

MISC LFSR CONFIGURATION REGISTER (MISC_CONFIG_LFSR, OFFSET 0x824)

The LFSR generator is used as a test/stress tool to verify that all blocks are of robust design and can tolerate extreme circumstances. When enabled, the LSFR value will be masked with the lfsr_mask value and then XORed into the final enable value. This allows specific state machines to be modulated while others may be controlled by the enable/disable bits above.

Table 128: MISC LFSR Configuration Register (misc_config_lfsr, Offset 0x824)

Bit	Name	Description	Mode	Reset
31–16	RESERVED		RO	0
15–0	DIV	This field controls the pre-scalar divider that is used as the time base for the LFSR enable generator. Each time the divisor (running on CORE_CLK) reaches this value, the LFSR will advance by one.	RW	0xff

MISC LFSR MASK REGISTER (MISC_LFSR_MASK_BITS, OFFSET 0x828)

Each bit in this register enables the LSFR value to be XORed into the final enable output. When all these bits are 0, the LSFR function is disabled.



Note: In BCM5709, All bits are SPLIT except bits 26 and 24 which are OR.

Table 129: MISC LFSR Mask Register (misc_lfsr_mask_bits, Offset 0x828)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	—	—
31–29	RESERVED		RO	0	—	—	X	X
28	RV2P_CMD_SCHEDULER_ENABLE	This bit controls or indicates status for the Command Scheduler block independently from all other blocks.	RO	0	—	—	X	X
27	RESERVED		RO	0	—	—	X	X
27	UMP_ENABLE	This bit controls or indicates status for the UMP sub-block of the Management processor.	RW	0	X	X	—	—
26	DMA_ENGINE_ENABLE	This bit controls or indicates status for the DMA Engine block independently from all other blocks. [OR]	RW	0	X	X	X	X
25	TIMER_ENABLE	This bit controls or indicates status for the Timer block independently from all other blocks.	RW	0	X	X	X	X
24	MGMT_PROCESSOR_ENABLE	This bit controls or indicates status for the Management Processor block independently from all other blocks. [OR]	RW	0	X	X	X	X
23	CMD_PROCESSOR_ENABLE	This bit controls or indicates status for the Command Processor block independently from all other blocks.	RW	0	X	X	X	X

Table 129: MISC LFSR Mask Register (misc_lfsr_mask_bits, Offset 0x828) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
22	CMD_SCHEDULER_ENABLE	This bit controls or indicates status for the Command Scheduler block independently from all other blocks.	RW	0	X	X	X	X
21	CONTEXT_ENABLE	This bit controls or indicates status for the Context block independently from all other blocks.	RW	0	X	X	X	X
20	MAILBOX_QUEUE_ENABLE	This bit controls or indicates status for the Mailbox Queue block independently from all other blocks.	RW	0	X	X	X	X
19	HOST_COALESCE_ENABLE	This bit controls or indicates status for the Host Coalesce block independently from all other blocks.	RW	0	X	X	X	X
18	COMPLETION_ENABLE	This bit controls or indicates status for the Completion block independently from all other blocks.	RW	0	X	X	X	X
17	RX_DMA_ENABLE	This bit controls or indicates status for the RX DMA block independently from all other blocks.	RW	0	X	X	X	X
16	RX_BD_CACHE_ENABLE	This bit controls or indicates status for the RX BD Cache block independently from all other blocks.	RW	0	X	X	X	X
15	RX_V2P_ENABLE	This bit controls or indicates status for the RX V2P block independently from all other blocks.	RW	0	X	X	X	X
14	RX_PROCESSOR_ENABLE	This bit controls or indicates status for the RX Processor block independently from all other blocks.	RW	0	X	X	X	X
13	RX_LOOKUP_ENABLE	This bit controls or indicates status for the RX Lookup block independently from all other blocks.	RW	0	X	X	X	X
12	RX_MBUF_ENABLE	This bit controls or indicates status for the RX MBUF block independently from all other blocks.	RW	0	X	X	X	X
11	RX_PARSER_CATCHUP_ENA BLE	This bit controls or indicates status for the RX Catchup Parser block independently from all other blocks.	RW	0	X	X	X	X
10	RX_PARSER_MAC_ENABLE	This bit controls or indicates status for the RX MAC Parser block independently from all other blocks.	RW	0	X	X	X	X
9	EMAC_ENABLE	This bit controls or indicates status for the EMAC block independently from all other blocks.	RW	0	X	X	X	X
8	TX_ASSEMBLER_ENABLE	This bit controls or indicates status for the TX Assembler block independently from all other blocks.	RW	0	X	X	X	X
7	TX_HEADER_Q_ENABLE	This bit controls or indicates status for the TX Header Queue sub-block of the TX Assembler independently from all other blocks.	RW	0	X	X	X	X

Table 129: MISC LFSR Mask Register (*misc_lfsr_mask_bits*, Offset 0x828) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
6	TX_PAYLOAD_Q_ENABLE	This bit controls or indicates status for the TX Payload Queue sub-block of the TX Assembler independently from all other blocks.	RW	0	X	X	X	X
5	TX_PATCHUP_ENABLE	This bit controls or indicates status for the TX Patchup block independently from all other blocks.	RW	0	X	X	X	X
4	TX_DMA_ENABLE	This bit controls or indicates status for the TX DMA block independently from all other blocks.	RW	0	X	X	X	X
3	TX_PROCESSOR_ENABLE	This bit controls or indicates status for the TX Processor block independently from all other blocks.	RW	0	X	X	X	X
2	TX_BD_CACHE_ENABLE	This bit controls or indicates status for the TX BD Cache block independently from all other blocks.	RW	0	X	X	X	X
1	TX_BD_READ_ENABLE	This bit controls or indicates status for the TX BD Read block independently from all other blocks.	RW	0	X	X	X	X
0	TX_SCHEDULER_ENABLE	This bit controls or indicates status for the TX Scheduler block independently from all other blocks.	RW	0	X	X	X	X

MISC ARBITRATION REQUEST REGISTERS (MISC_ARB_REQ[5], OFFSET 0x82C)

Description	Mode	Reset	06	08	09	16
These registers are the write commands to request a resource from the generic SC software arbiter implemented here. By writing one of these bits as a 1, that resource is requested for a particular user. Writing 0's to any of these bits has no effect. The bits are organized such that each register address is a different user and each bit position represents a different generic arbiter channel. This means that the arbiter supports 32 channels with 5 users for each channel.		0	X	X	X	X

Example: If you are user #4 and want to request resource number 23, then set bit 23 of arb_req[4]. After requesting the resource, poll the appropriate arb_gnt register until your user level is indicated.

MISC ARBITRATION FREE REGISTERS (MISC_ARB_FREE[5], OFFSET 0x840)

Description	Mode	Reset	06	08	09	16
These registers are the write commands to free a resource from the generic SC software arbiter implemented here. By writing one of these bits as a 1, that resource is freed for a particular user. Writing 0's to any of these bits has no effect.	0	X X X X				

MISC ARBITRATION REQUEST STATUS REGISTERS (MISC_ARB_REQ_STATUS[5], OFFSET 0x854)

Description	Mode	Reset	06	08	09	16
These registers allow the current state of request for each user to be read for RO debugging reasons. The bits are organized such that each register address is a different user and each bit position represents a different generic arbiter channel.	0	X X X X				

MISC ARBITRATION GRANT 0 REGISTER (MISC_ARB_GNT0, OFFSET 0x868)*Table 130: MISC Arbitration Grant 0 Register (misc_arb_gnt0, Offset 0x868)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30–28	7	Arbiter status for channel 7. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
27	RESERVED		RO	0	X	X	X	X
26–24	6	Arbiter status for channel 6. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
23	RESERVED		RO	0	X	X	X	X
22–20	5	Arbiter status for channel 5. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
19	RESERVED		RO	0	X	X	X	X
18–16	4	Arbiter status for channel 4. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
15	RESERVED		RO	0	X	X	X	X
14–12	3	Arbiter status for channel 3. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
11	RESERVED		RO	0	X	X	X	X
10–8	2	Arbiter status for channel 2. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
7	RESERVED		RO	0				
6–4	1	Arbiter status for channel 1. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
3	RESERVED		RO	0	X	X	X	X
2–0	0	Arbiter status for channel 0. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X

MISC ARBITRATION GRANT 1 REGISTER (MISC_ARB_GNT1, OFFSET 0x86C)*Table 131: MISC Arbitration Grant 1 Register (misc_arb_gnt1, Offset 0x86c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30–28	15	Arbiter status for channel 15. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
27	RESERVED		RO	0	X	X	X	X
26–24	14	Arbiter status for channel 14. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
23	RESERVED		RO	0	X	X	X	X
22–20	13	Arbiter status for channel 13. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
19	RESERVED		RO	0	X	X	X	X
18–16	12	Arbiter status for channel 12. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
15	RESERVED		RO	0	X	X	X	X
14–12	11	Arbiter status for channel 11. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
11	RESERVED		RO	0	X	X	X	X
10–8	10	Arbiter status for channel 10. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
7	RESERVED		RO	0	X	X	X	X
6–4	9	Arbiter status for channel 9. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
3	RESERVED		RO	0	X	X	X	X
2–0	8	Arbiter status for channel 8. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X

MISC ARBITRATION GRANT 2 REGISTER (MISC_ARB_GNT2, 0x870)*Table 132: MISC Arbitration Grant 2 Register (misc_arb_gnt2, 0x870)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30–28	23	Arbiter status for channel 23. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
27	RESERVED		RO	0	X	X	X	X
26–24	22	Arbiter status for channel 22. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
23	RESERVED		RO	0	X	X	X	X
22–20	21	Arbiter status for channel 21. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
19	RESERVED		RO	0	X	X	X	X
18–16	20	Arbiter status for channel 20. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
15	RESERVED		RO	0	X	X	X	X
14–12	19	Arbiter status for channel 19. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
11	RESERVED		RO	0	X	X	X	X
10–8	18	Arbiter status for channel 18. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
7	RESERVED		RO	0	X	X	X	X
6–4	17	Arbiter status for channel 17. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
3	RESERVED		RO	0	X	X	X	X
2–0	16	Arbiter status for channel 16. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X

MISC ARBITRATION GRANT 3 REGISTER (MISC_ARB_GNT3, OFFSET 0x874)*Table 133: MISC Arbitration Grant 3 Register (misc_arb_gnt3, Offset 0x874)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30–28	31	Arbiter status for channel 31. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
27	RESERVED		RO	0	X	X	X	X
26–24	30	Arbiter status for channel 30. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
23	RESERVED		RO	0	X	X	X	X
22–20	29	Arbiter status for channel 29. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
19	RESERVED		RO	0	X	X	X	X
18–16	28	Arbiter status for channel 28. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
15	RESERVED		RO	0	X	X	X	X
14–12	27	Arbiter status for channel 27. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
11	RESERVED		RO	0	X	X	X	X
10–8	26	Arbiter status for channel 26. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
7	RESERVED		RO	0	X	X	X	X
6–4	25	Arbiter status for channel 25. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X
3	RESERVED		RO	0	X	X	X	X
2–0	24	Arbiter status for channel 24. The value reflects the lowest number of all the current requesters currently requesting. This value is 0x7 when no requestor is active.	RO	0x7	X	X	X	X

MISC PRBS CONTROL REGISTER (MISC_PRBS_CONTROL, OFFSET 0x878)

This register controls the pseudo random BIST logic for the SerDes block.

Table 134: MISC PRBS Control Register (misc_prbs_control, Offset 0x878—BCM5706 Only)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–6	RESERVED		RO	0	X	—	—	—
31–0	RESERVED		RO	0	—	X	X	X
5–4	ORDER	This value sets the polynomial.	RW	0	X	—	—	—
		Value Name Description						
		0 7 TH $x(n) = 1 + x(6) + x(7)$						
		1 15 TH $x(n) = 1 + x(14) + x(15)$						
		2 23 RD $x(n) = 1 + x(18) + x(23)$						
		3 31 ST $x(n) = 1 + x(28) + x(31)$						
3	ERR_CLR	When this bit is written as a 1, the error counter will be held in the cleared state. This bit must be 0 to allow the error counter to count.	RW	0	X	—	—	—
2	INV	When this bit is 1, an inverted pattern will be used. When this bit is 0, a non-inverted pattern will be used.	RW	0	X	—	—	—
1	RSTB	This bit must be set to 1 to allow the PRBS generator/monitor to run. Setting this bit to 0 will hold the PRBS generator and monitor in reset.	RW	0	X	—	—	—
0	EN	When set to 1, the PRBS generator and monitor will run. The INV and ORDER bits must be configured before this bit is set to 1.	RW	0	X	—	—	—

MISC PRBS STATUS REGISTER (MISC_PRBS_STATUS, OFFSET 0x87C)

This register provides visibility into the status of the SerDes block.

Table 135: MISC PRBS Status Register (misc_prbs_status, Offset 0x87C—BCM5706 Only)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–20	RESERVED		RO	0	X	—	—	—
31–0	RESERVED		RO	0	—	X	X	X
19–16	STATE	—	RO	0	X	—	—	—
15–2	ERRORS	This register returns the number of errors that have been encountered. This count is cleared to 0 when the ERR_CLR bit is 1.	RO	0	X	—	—	—
1	STKY	This bit is set each time the LOCK bit transitions from 1 to 0 so that loss of lock of a period of operation can be detected. This bit is cleared to 0 by the ERR_CLR bit.	RO	0	X	—	—	—
0	LOCK	This bit indicates that the PRBS Monitor has locked onto the generator when it is 1.	RO	0	X	—	—	—

MISC SMBus/ASF CONTROL REGISTER (MISC_SM ASF_CONTROL, OFFSET 0x880)**Table 136: MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	SMB_EARLY_ATTN	When this bit is 1, the SMBus interface sets the SMB_ATTN bit as soon as slave activity is detected. When this bit is 0, the SMB_ATTN bit is not set until an address match occurs.	RW	0	X	X	-	-
31–0	RESERVED		RO	0	-	-	X	X
30	EN_NIC_SMB_ADDR_0	When this bit is 1, a value of 0 is the third of three SMBus addresses which will be used to match for incoming messages when the SMB_ADDR_FILT bit is 1.	RW	0	X	X	-	-
29–24	NIC_SMB_ADDR2	This is the second of three SMBus addresses that will be used to match for incoming messages when the SMB_ADDR_FILT bit is 1. When the SMB_ADDR_FILT bit is 0, this value is ignored.	RW	0	X	X	-	-
23–22	RESERVED		RO	0	X	X	-	-
21–16	NIC_SMB_ADDR1	This is the first of three SMBus addresses that will be used to match for incoming messages when the SMB_ADDR_FILT bit is 1. When the SMB_ADDR_FILT bit is 0, this value is ignored.	RW	0	X	X	-	-
15	SMB_AUTOREAD	When this bit is 1, the SMB_IN_RDY bit will clear automatically whenever the "PCI Clock Control Register (pcicfg_pci_clock_control_bits, Offset 0x70)" on page 174 is read.	RW	0	X	X	-	-
14	SMB_NO_ADDR_FILT	When this bit is 0, the SMB_EVENT field will only be asserted if the incoming SMBus message matches either the address in the NIC_SMB_ADDR1 or NIC_SMB_ADDR2 fields, or the SMB_EVENT field will be cleared if the EN_NIC_SMB_ADDR_0 bit is 1.	RW	0	X	X	-	-
13	SMB_BB_EN	When this bit is 1, the SMBus block is placed into bit-bang mode.	RW	0	X	X	-	-
12	SMB_EN	When this bit is 1, the SMBus block is enabled for operation.	RW	0	X	X	-	-
11–8	RESERVED		RW	0	X	X	-	-
7	SMB_EVENT	This bit changes to 1 each time the SMBus state machine receives a message. Writing a 1 to this position will clear this bit. When this bit is 1, the SMB0_EVENT bit will be 1 in each processor.	WC	0	X	X	-	-

Table 136: MISC SMBus/ASF Control Register (*misc_sm_asf_control*, Offset 0x880) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
6	RT_TO	This bit changes to 1 each time the RETRANSMIT timer reaches 0. Writing a 1 to this position will clear this bit. When this bit is 1, the SMB0_EVENT bit will be 1 in each processor.	WC	0	X	X	-	-
5	PL_TO	This bit changes to 1 each time the POLL_LEGACY timer reaches 0. Writing a 1 to this position will clear this bit. When this bit is 1, the SMB0_EVENT bit will be 1 in each processor.	WC	0	X	X	-	-
4	PA_TO	This bit changes to 1 each time the POLL ASF timer reaches 0. Writing a 1 to this position will clear this bit. When this bit is 1, the SMB0_EVENT bit will be 1 in each processor.	WC	0	X	X	-	-
3	HB_TO	This bit changes to 1 each time the HEARTBEAT timer reaches 0. Writing a 1 to this position will clear this bit. When this bit is 1, the SMB0_EVENT bit will be 1 in each processor.	WC	0	X	X	-	-
2	WG_TO	This bit changes to 1 each time the WATCHDOG timer reaches 0. Writing a 1 to this position will clear this bit. When this bit is 1, the SMB0_EVENT bit will be 1 in each processor.	WC	0	X	X	-	-
1	TSC_EN	When this bit is 1, the timestamp counter register counts by 1 each second. When this bit is 0 the timestamp register holds its value.	RW	0	X	X	-	-
0	ASF_RST	When this bit is written as 1, the ASF and SMBUS blocks are reset. Writing this bit as 0 has no effect.	SC	0	X	X	-	-

MISC SMBus IN REGISTER (MISC_SMB_IN, OFFSET 0x884)*Table 137: MISC SMBus In Register (misc_smb_in, Offset 0x884)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–14	RESERVED		RO	0	X	X	X	X
13–11	STATUS	This value is set each time the DONE bit is set to 1 and shows the status of the preceding transfer.	RW	0	X	X	—	—
		Value	Name	Description				
		0x0	OK	The transfer reception completed without error.				
		0x1	PEC	The transfer had a PEC error during reception.				
		0x2	OFLOW	The transfer had a FIFO overflow during the reception.				
		0x3	STOP	The SMBus stopped unexpectedly during the reception.				
		0x4	TIMEOUT	The SMBus timed out during the reception.				
13–0	RESERVED		RO	0	—	—	X	X
10	FIRSTBYTE	This bit is 1 when the DAT_IN value is valid and it is the first byte of a transfer.	RW	0	X	X	—	—
9	DONE	This bit is 1 when the DAT_IN value is valid and it is the last byte of a transfer.	RW	0	X	X	—	—
8	RDY	This bit is 1 when the DAT_IN value is valid after reception of a byte message.	RW	0	X	X	—	—
7–0	DAT_IN	This value is the incoming data byte from the SMBus controller. This value is valid with the RDY bit is 1.	RW	0	X	X	—	—

MISC SMBus OUT REGISTER (MISC_SMB_OUT, OFFSET 0x888)*Table 138: MISC SMBus Out Register (misc_smb_out, Offset 0x888)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–29	RESERVED		RO	0	X	X	X	X
28	SMB_OUT_CLK_IN	This bit reflects the current input value of the CLK pin. When this CLK pin is high, this bit will read as 1. When the CLK pin is low, this pin will read as 0.	RO	0	X	X	–	–
28–0	RESERVED		RO	0	–	–	X	X
27	SMB_OUT_CLK_EN	When the SMBus interface is configured for bit-bang mode, this bit controls the output enable for the CLK pin. When this bit is 0, the CLK pin will drive low. When this bit is 1, the CLK pin will float.	RW	0	X	X	–	–
26	SMB_OUT_DAT_IN	This bit reflects the current input value of the DAT pin. When this DAT pin is high, this bit will read as 1. When the DAT pin is low, this pin will read as 0.	RO	0	X	X	–	–
25	SMB_OUT_DAT_EN	When the SMBus interface is configured for bit-bang mode, this bit controls the output enable for the DAT pin. When this bit is 0, the DAT pin will drive low. When this bit is 1, the DAT pin will float.	RW	0	X	X	–	–
24	SMB_OUT_SLAVEMODE	When this bit set, the SMBus interface will operate in slave mode rather than master mode.	RW	0	X	X	–	–

Table 138: MISC SMBus Out Register (*misc_smb_out*, Offset 0x888) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
23–20	SMB_OUT_STATUS	This value is set when the SMB_OUT_START bit is cleared with the encoded status of the previous transfer.	RW	0	X	X	–	–
		Val Name Description						
	0x0	OK	Transmission is OK.					
	0x1	FIRST_N ACK	SMBus was NACKed on the first byte of the transmission.					
	0x2	UFLOW	SMBus interface had a output FIFO under-run during the transmission.					
	0x3	STOP	SMBus stopped unexpectedly during the transmission.					
	0x4	TIMEOU T	SMBus timed out during the transmission.					
	0x5	FIRST_L OST	SMBus master lost arbitration during the first byte of the transmission.					
	0x6	BADACK	SMBus master ACKed on what should have been the last byte of the transfer.					
	0x9	SUB_NA CK	SMBus was NACKed on a subsequent byte of the transmission.					
	0xd	SUB_LO ST	SMBus master lost arbitration during a subsequent byte of the transmission.					
19–14	SMB_READ_LEN	This value sets the number of bytes in the read portion of the transaction if the GET_RX_LEN bit is 0.	RW	0	X	X	–	–
13	GET_RX_LEN	When this bit is set to 1, The RX length is taken from the first byte of the read data. When this bit is 0, the SMB_READ_LEN field is used for the RX length.	RW	0	X	X	–	–
12	ENB_PEC	When this bit is set to 1, the packet error check byte is enabled.	RW	0	X	X	–	–
11	ACC_TYPE	When this bit is set to 1, the send is executed as a Read command.	RW	0	X	X	–	–
10	LAST	When this bit and the RDY bit are written as 1, the byte in DAT_OUT will be transmitted as the end of an SMBus message.	RW	0	X	X	–	–
9	START	When this bit and the RDY bit are written as 1, the byte in DAT_OUT will be transmitted as the start of an SMBus message. The bit will self clear when the send is complete. This bit can be manually cleared by writing a 1 to the bit position.	SC	0	X	X	–	–

Table 138: MISC SMBus Out Register (misc_smb_out, Offset 0x888) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
8	RDY	This bit must be written as 1 with the DAT_OUT value is written and will self-clear when the data bytes is transferred into the internal FIFO.	SC	0	X	X	-	-
7–0	DAT_OUT	This value is the outgoing data byte to the SMBus controller. Value is accepted when RDY is written as 1.	RW	0	X	X	-	-

MISC SMBus WATCHDOG REGISTER (MISC_SMB_WATCHDOG, OFFSET 0x88C)**Table 139: MISC SMBus Watchdog Register (misc_smb_watchdog, Offset 0x88c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	-	-
15–0	WATCHDOG	This value counts down to 0 once each second and sets the WG_TO bit of the "MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)" on page 251 when it reaches 0. The counter stops when it reaches 0.	RW	0	X	X	-	-
15–0	RESERVED		RO	0	-	-	X	X

MISC SMBus HEARTBEAT REGISTER (MISC_SMB_HEARTBEAT, OFFSET 0x890)**Table 140: MISC SMBus Heartbeat Register (misc_smb_heartbeat, Offset 0x890)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–0	HEARTBEAT	This value counts down to 0 once each second and sets the HB_TO bit of the "MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)" on page 251 when it reaches 0. The counter stops when it reaches 0.	RW	0	X	X	-	-
15–0	RESERVED		RO	0	-	-	X	X

MISC SMBus Poll ASF Register (MISC_SMB_POLL ASF, Offset 0x894)*Table 141: MISC SMBus Poll ASF Register (misc_smb_poll_asf, Offset 0x894)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–0	POLL ASF	This value counts down to 0 once each 5 msec. and sets the PA_TO bit of the “MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)” on page 251 when it reaches 0. The counter stops when it reaches 0.	RW	0	X	X	—	—
15–0	RESERVED		RO	0	—	—	X	X

MISC SMBus Poll Legacy Register (MISC_SMB_POLL_LEGACY, Offset 0x898)*Table 142: MISC SMBus Poll Legacy Register (misc_smb_poll_legacy, Offset 0x898)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–0	POLL_LEGACY	This value counts down to 0 once each 250 msec. and sets the PL_TO bit of the “MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)” on page 251 when it reaches 0. The counter stops when it reaches 0.	RW	0	X	X	—	—
15–0	RESERVED		RO	0	—	—	X	X

MISC SMBus Retransmit Register (MISC_SMB_RETRAN, Offset 0x89C)*Table 143: MISC SMBus Retransmit Register (misc_smb_retran, Offset 0x89c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–8	RESERVED		RO	0	X	X	X	X
7–0	RETRAN	This value counts down to 0 once each second and sets the RT_TO bit of the “MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)” on page 251 when it reaches 0. The counter stops when it reaches 0.	RW	0	X	X	—	—
7–0	RESERVED		RO	0	—	—	X	X

MISC SMBus TIMESTAMP REGISTER (MISC_SMB_TIMESTAMP, OFFSET 0x8A0)*Table 144: MISC SMBus Timestamp Register (misc_smb_timestamp, Offset 0x8a0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	TIMESTAMP	This value counts up once each second and rolls to 0 each time it passes 0xffffffff. This counter only counts when the TSC_EN bit of "MISC SMBus/ASF Control Register (misc_sm_asf_control, Offset 0x880)" on page 251 is 1.	RW	0	X	X	–	–

MISC PERR ENABLE 0 REGISTER (MISC_PERR_ENA0, OFFSET 0x8A4)*Table 145: MISC PERR Enable 0 Register (misc_perr_ena0, Offset 0x8a4)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RV2P_MISC_CB0REGS	–	RW	0	X	X	–	–
31	TSCH_LR_PERR_EN	Parity mask enable	RW	0	–	–	X	X
30	RDE_MISC_RPM	–	RW	0	X	X	–	–
30	TBDC_PERR_EN	Parity mask enable	RW	0	–	–	X	X
29	RDE_MISC_RPC	–	RW	0	X	X	–	–
29	TDMA_PERR_EN	Parity mask enable	RW	0	–	–	X	X
28	RBUF_MISC_PTR	–	RW	0	X	X	–	–
28	THBUF_PERR_EN	Parity mask enable	RW	0	–	–	X	X
27	RBUF_MISC_MB	–	RW	0	X	X	–	–
27	TPBUF_PERR_EN	Parity mask enable	RW	0	–	–	X	X
26	RBDC_MISC	–	RW	0	X	X	–	–
26	RV2P_CB0REGS_PERR_EN	Parity mask enable	RW	0	–	–	X	X
25	MQ_MISC_CTX	–	RW	0	X	X	–	–
25	RV2P_CB1REGS_PERR_EN	Parity mask enable	RW	0	–	–	X	X
24	MCP_MISC_SCPAD	–	RW	0	X	X	–	–
24	RV2P_P1IRAM_PERR_EN	Parity mask enable	RW	0	–	–	X	X
23	MCP_MISC_REGF	–	RW	0	X	X	–	–
23	RV2P_P2IRAM_PERR_EN	Parity mask enable	RW	0	–	–	X	X
22	HC_MISC_DMA	–	RW	0	X	X	–	–
22	RBUF_DATAMEM_PERR_EN	Parity mask enable	RW	0	–	–	X	X
21	DMAE_MISC_DW2	–	RW	0	X	X	–	–
21	RBUF_PTRMEM_PERR_EN	Parity mask enable	RW	0	–	–	X	–
20	DMAE_MISC_DW1	–	RW	0	X	X	–	–
20	RPC_DFIFOMEM_PERR_EN	Parity mask enable	RW	0	–	–	X	X
19	DMAE_MISC_DW0	–	RW	0	X	X	–	–
19	RPM_DFIFOMEM_PERR_EN	Parity mask enable	RW	0	–	–	X	X
18	DMAE_MISC_DR4	–	RW	0	X	X	–	–
18	MQ_CTX_PERR_EN	Parity mask enable	RW	0	–	–	X	X



Table 145: MISC PERR Enable 0 Register (misc_perr_ena0, Offset 0x8a4) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
17	DMAE_MISC_DR3	–	RW	0	X	X	–	–
17	CS_TMEM_PERR_EN	Parity mask enable	RW	0	–	–	X	X
16	DMAE_MISC_DR2	–	RW	0	X	X	–	–
16	TXP_SCPAD_PERR_EN	Parity mask enable	RW	0	–	–	X	X
15	DMAE_MISC_DR1	–	RW	0	X	X	–	–
15	TXP_CTXC_PERR_EN	Parity mask enable	RW	0	–	–	X	
14	DMAE_MISC_DR0	–	RW	0	X	X	–	–
14	TPAT_SCPAD_PERR_EN	Parity mask enable	RW	0	–	–	X	X
13	CTX_MISC_PGTBL	–	RW	0	X	X	–	–
13	RXP_SCPAD_PERR_EN	Parity mask enable	RW	0	–	–	X	X
12	CTX_MISC_ACCM5	–	RW	0	X	X	–	–
12	RXP_CTXC_PERR_EN	Parity mask enable	RW	0	–	–	X	X
11	CTX_MISC_ACCM4	–	RW	0	X	X	–	–
11	RXP_RBUFC_PERR_EN	Parity mask enable	RW	0	–	–	X	X
10	CTX_MISC_ACCM3	–	RW	0	X	X	–	–
10	CP_SCPAD_PERR_EN	Parity mask enable	RW	0	–	–	X	X
9	CTX_MISC_ACCM2	–	RW	0	X	X	–	–
9	CP_CTXC_PERR_EN	Parity mask enable	RW	0	–	–	X	X
8	CTX_MISC_ACCM1	–	RW	0	X	X	–	–
8	COM_SCPAD_PERR_EN	Parity mask enable	RW	0	–	–	X	X
7	CTX_MISC_ACCM0	–	RW	0	X	X	–	–
7	COM_CTXC_PERR_EN	Parity mask enable	RW	0	–	–	X	X
6	CS_MISC_TMEM	–	RW	0	X	X	–	–
6	CTX_MIRROR_PERR_EN	Parity mask enable	RW	0	–	–	X	X
5	CP_MISC_SCPAD	–	RW	0	X	X	–	–
5	CTX_CACHE_PERR_EN	Parity mask enable	RW	0	–	–	X	X
4	CP_MISC_REGF	–	RW	0	X	X	–	–
4	CTX_PGTBL_PERR_EN	Parity mask enable	RW	0	–	–	X	X
3	CP_MISC_CTXC	–	RW	0	X	X	–	–
3	CTX_USAGE_CNT_PERR_EN	Parity mask enable	RW	0	–	–	X	X
2	COM_MISC_SCPAD	–	RW	0	X	X	–	–
2	RPM_ACPIBEMEM_ERR_EN	Parity mask enable	RW	0	–	–	X	X
1	COM_MISC_REGF	–	RW	0	X	X	–	–
1	CP_DMAE_PERR_EN	Parity mask enable	RW	0	–	–	X	X
0	COM_MISC_CTXC	–	RW	0	X	X	–	–
0	COM_DMAE_PERR_EN	Parity mask enable	RW	0	–	–	X	X



Note: This version of register 0x8a4 applies to BCM5709 only.

MISC_PERR_ENABLE_1 REGISTER (MISC_PERR_ENA1, OFFSET 0x8A8)*Table 146: MISC_PERR_Enable_1 Register (misc_perr_ena1, Offset 0x8a8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RLUPQ_MISC	–	RW	0	X	X	–	–
31–30	RESERVED		RO	0	–	–	X	X
30	RXPCQ_MISC	–	RW	0	X	X	–	–
29	RXPQ_MISC	–	RW	0	X	X	–	–
29	MQ_IDX_PERR_EN	Parity mask enable	RW	0	–	–	X	X
28	RV2PTQ_MISC	–	RW	0	X	X	–	–
28	RV2PCSQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
27	RV2PPQ_MISC	–	RW	0	X	X	–	–
27	RV2PCS_TMEM	Parity mask enable	RW	0	–	–	X	X
26	RV2PMQ_MISC	–	RW	0	X	X	–	–
26	RLUP_CID_PERR_EN	Parity mask enable	RW	0	–	–	X	X
25	MCPQ_MISC	–	RW	0	X	X	–	–
25	CSQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
24	CPQ_MISC	–	RW	0	X	X	–	–
24	CPQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
23	CSQ_MISC	–	RW	0	X	X	–	–
23	RV2PMQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
22	RDMAQ_MISC	–	RW	0	X	X	–	–
22	RV2PTQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
21	UMP_MISC_TX	–	RW	0	X	X	–	–
21	RXPCQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
20	UMP_MISC_RX	–	RW	0	X	X	–	–
20	COMXQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
19	UMP_MISC_FIOTX	–	RW	0	X	X	–	–
19	TSCHQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
18	UMP_MISC_FIORX	–	RW	0	X	X	–	–
18	TBDRQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
17	TXP_MISC_SCPAD	–	RW	0	X	X	–	–
17	TASQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
16	TXP_MISC_REGF	–	RW	0	X	X	–	–
16	RDMAQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
15	TXP_MISC_CTXC	–	RW	0	X	X	–	–
15	RV2PPQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
14	TSCH_MISC_LR	–	RW	0	X	X	–	–
14	RXPQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
13	TPBUF_MISC_MB	–	RW	0	X	X	–	–
13	RLUPQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
12	TPAT_MISC_SCPAD	–	RW	0	X	X	–	–
12	COMQ_PERR_EN	Parity mask enable	RW	0	–	–	X	X
11	TPAT_MISC_REGF	–	RW	0	X	X	–	–



Table 146: MISC PERR Enable 1 Register (misc_perr_ena1, Offset 0x8a8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11	COMTQ_PERR_EN	Parity mask enable	RW	0	—	—	X	X
10	THBUF_MISC_MB1	—	RW	0	X	X	—	—
10	TXPQ_PERR_EN	Parity mask enable	RW	0	—	—	X	X
9	THBUF_MISC_MB0	—	RW	0	X	X	—	—
9	TDMAQ_PERR_EN	Parity mask enable	RW	0	—	—	X	X
8	TDMA_MISC	—	RW	0	X	X	—	—
8	MCPQ_PERR_EN	Parity mask enable	RW	0	—	—	X	X
7	TBDC_MISC	—	RW	0	X	X	—	—
7	TPATQ_PERR_EN	Parity mask enable	RW	0	—	—	X	X
6	RXP_MISC_RBUFC	—	RW	0	X	X	—	—
6	HC_CONSUMSTB	Parity mask enable	RW	0	—	—	X	X
5	RXP_MISC_SCPAD	—	RW	0	X	X	—	—
5	HC_PRODUCSTB	Parity mask enable	RW	0	—	—	X	X
4	RXP_MISC_REGF	—	RW	0	X	X	—	—
4	HC_MSIX_PERR_EN	Parity mask enable	RW	0	—	—	X	X
3	RXP_MISC_CTXC	—	RW	0	X	X	—	—
3	HC_STATS_PERR_EN	Parity mask enable	RW	0	—	—	X	X
2	RV2P_MISC_P2IRAM	—	RW	0	X	X	—	—
2	RDMA_DFIFO	Parity mask enable	RW	0	—	—	X	X
1	RV2P_MISC_P1IRAM	—	RW	0	X	X	—	—
1	RESERVED	—	RO	0	—	—	X	X
0	RV2P_MISC_CB1REGS	—	RW	0	X	X	—	—
0	RBDC_PERR_EN	Parity mask enable	RW	0	—	—	X	X

MISC PERR ENABLE 2 REGISTER (MISC_PERR_ENA2, OFFSET 0x8AC)**Table 147: MISC PERR Enable 2 Register (misc_perr_ena2, Offset 0x8ac)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–9	RESERVED	—	RO	0	X	X	X	X
8	TASQ_MISC	—	RW	0	X	X	—	—
8	RESERVED	—	RW	0	—	—	X	X
7	TPATQ_MISC	—	RW	0	X	X	—	—
7	RESERVED	—	RW	0	—	—	X	X
6	TDMAQ_MISC	—	RW	0	X	X	—	—
6	PCIE REPLAY_PERR_EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X
5	TXPQ_MISC	—	RW	0	X	X	—	—
5	HB_MEM_PERR_EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X
4	TBDRQ_MISC	—	RW	0	X	X	—	—
4	MCP_SCPAD_PERR_EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X
3	TSCHQ_MISC	—	RW	0	X	X	—	—
3	MCP_ROM_PERR_EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X

Table 147: MISC PERR Enable 2 Register (misc_perr_ena2, Offset 0x8ac) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2	COMTQ_MISC	—	RW	0	X	X	—	—
2	UMP RX PERR EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X
1	COMXQ_MISC	—	RW	0	X	X	—	—
1	UMP TX PERR EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X
0	COMQ_MISC	—	RW	0	X	X	—	—
0	TGT FIFO PERR EN	Parity mask enable [OR]	RW	0:CMN	—	—	X	X

MISC VOLTAGE REGULATOR CONTROL REGISTER (MISC_VREG_CONTROL, OFFSET 0x8B4)**Table 148: MISC Voltage Regulator Control Register (misc_vreg_control, Offset 0x8b4)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–12	RESERVED		RO	0	X	X	X	X
11–8	RESERVED		RO	0	X	X	—	—
11–8	1_0 MGMT	SHARE:Control for the MGMT 1.0V regulator.	RW	0x7:Hard	—	—	X	X
Value	Name	Description						
0	PLUS14	Set to +14%;						
1	PLUS12	Set to +12%;						
2	PLUS10	Set to +10%;						
3	PLUS8	Set to +8%;						
4	PLUS6	Set to +6%;						
5	PLUS4	Set to +4%;						
6	PLUS2	Set to +2%;						
7	NOM	Set to 0%, this is nominal and default value						
8	MINUS2	Set to -2%;						
9	MINUS4	Set to -4%;						
10	MINUS6	Set to -6%;						
11	MINUS8	Set to -8%;						
12	MINUS10	Set to -10%;						
13	MINUS12	Set to -12%;						
14	MINUS14	Set to -14%;						
15	MINUS16	Set to -16%						
7–4	2_5	—	RW	0	X	X	—	—

Table 148: MISC Voltage Regulator Control Register (misc_vreg_control, Offset 0x8b4) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7-4	2_5	Control for the 2.5V regulator. [SHARE]	RW	0x7:HW	-	-	X	X
	Value	Name	Description					
	0	PLUS14	Set to +14%;					
	1	PLUS12	Set to +12%;					
	2	PLUS10	Set to +10%;					
	3	PLUS8	Set to +8%;					
	4	PLUS6	Set to +6%;					
	5	PLUS4	Set to +4%;					
	6	PLUS2	Set to +2%;					
	7	NOM	Set to 0%, this is nominal and default value					
	8	MINUS2	Set to -2%;					
	9	MINUS4	Set to -4%;					
	10	MINUS6	Set to +6%;					
	11	MINUS8	Set to -8%;					
	12	MINUS10	Set to -10%;					
	13	MINUS12	Set to -12%;					
	14	MINUS14	Set to -14%;					
	15	MINUS16	Set to -16%					
3-0	1_2	-	RW	0	X	X	-	-
3-0	1_0_MAIN	SHARE:Control for the MAIN 1.0V regulator.	RW	0x7:Hard	-	-	X	X
	Value	Name	Description					
	0	PLUS14	Set to +14%;					
	1	PLUS12	Set to +12%;					
	2	PLUS10	Set to +10%;					
	3	PLUS8	Set to +8%;					
	4	PLUS6	Set to +6%;					
	5	PLUS4	Set to +4%;					
	6	PLUS2	Set to +2%;					
	7	NOM	Set to 0%, this is nominal and default value					
	8	MINUS2	Set to -2%;					
	9	MINUS4	Set to -4%;					
	10	MINUS6	Set to +6%;					
	11	MINUS8	Set to -8%;					
	12	MINUS10	Set to -10%;					
	13	MINUS12	Set to -12%;					
	14	MINUS14	Set to -14%;					
	15	MINUS16	Set to -16%					

MISC FINAL CLOCK CONTROL VALUE REGISTER (MISC_VREG_CONTROL, OFFSET 0x8B8)*Table 149: MISC Final Clock Control Value Register (misc_vreg_control, Offset 0x8b8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–6	MISC_FINAL_CLK_CTRL_VAL	This field displays the value actually controlling the clock blocks.	RO	0	X	X	–	–
31–6	RESERVED		RO	0	–	–	X	X
5–0	RESERVED		RO	0	X	X	X	X

GENERAL-PURPOSE HARDWARE CONTROL 0 REGISTER (GP_HW_CTL0, OFFSET 0x8BC)*Table 150: General-Purpose Hardware Control 0 Register (gp_hw_ctl0, Offset 0x8bc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–4	RESERVED		RO	0	X	–	–	–
31–30	OSCCTRL_XTAL_ADJ	Bit OSCCTRL[47–46] of the 25-MHz oscillator. Controls xtal_adjCM[1–0], adjust the common mode of the oscillator. [SHARE]	RW	0	–	X	X	X
		Value Name Description						
		0 1P57 1.57V						
		1 1P45 1.45V						
		2 1P62 1.62V						
		3 1P66 1.66V						
29–28	OSCCTRL_ICBUF_ADJ	Bit OSCCTRL[45–44] of the 25-MHz oscillator. Controls lcbuf_adj[1–0], adjust the current to CMOS to CML buffer. [SHARE]	RW	0	–	X	X	X
		Value Name Description						
		0 240 UA 240 μ A						
		1 160 UA 160 μ A						
		2 400 UA 400 μ A						
		3 320 UA 320 μ A						
27–26	OSCCTRL_IAMP_ADJ	Bit OSCCTRL[43–42] of the 25-MHz oscillator. Controls lamp_adj[1–0], adjust the current of gain amplifier. [SHARE]	RW	0	–	X	X	X
		Value Name Description						
		0 240 UA 240 μ A						
		1 160 UA 160 μ A						
		2 400 UA 400 μ A						
		3 320 UA 320 μ A						

Table 150: General-Purpose Hardware Control 0 Register (*gp_hw_ctl0*, Offset 0x8bc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25–24	OSCCTRL_PTAT	Bit OSCCTRL[39–38] of the 25-MHz oscillator. Controls PTAT value. [SHARE]	RW	0x1	–	X	X	X
		Value Name Description						
		0x0 M6P –6%.						
		0x1 M0P 0%.						
		0x2 P0P 0%.						
		0x3 P6P +6%.						
23–22	OSCCTRL_CTAT	Bit OSCCTRL[37–36] of the 25-MHz oscillator. Controls CTAT value. [SHARE]	RW	0x1	–	X	X	X
		Value Name Description						
		0x0 M6P –6%.						
		0x1 M0P 0%.						
		0x2 P0P 0%.						
		0x3 P6P +6%.						
21	OSCCTRL_PRE1DIS	Bit OSCCTRL[35] of the 25-MHz oscillator. Enables first pre-driver. [SHARE]	RW	0	–	X	X	X
20	OSCCTRL_PRE2DIS	Bit OSCCTRL[34] of the 25-MHz oscillator. Disables second pre-driver for driver A, B, C, and D. [SHARE]	RW	0	–	X	X	X
19–16	OSCCTRL_DAI	Bits OSCCTRL[31–28] of the 25-MHz oscillator. [SHARE]	RW	0x3	–	X	X	X
		Value Name Description						
		0x0 3MA 3 mA						
		0x1 2P5MA 2.5 mA						
		0x3 2P0MA 2.0 mA						
		0x5 1P5MA 1.5 mA						
		0x7 1P0MA 1.0 mA						
		0xf PWRDN Power down						
15	PARALLEL_DETECT_DEF_OR	This bit controls the parallel detect def strap input of the SerDes block. [OR]	RW	0:Hard	–	X	X	X
14	AUTODETECT_DIS_DEF_OR	The inversion of this bit controls the autodetect def strap input of the SerDes block. [OR]	RW	0:Hard	–	X	X	X
13	FORCE2500_DEF_OR	This bit controls the force2500 def strap input of the SerDes block. [OR]	RW	0:Hard	–	X	X	X
12	FIBER_MODE_DIS_DEF_OR	The inversion of this bit controls the fiber mode def strap input of the SerDes block. [OR]	RW	0:Hard	–	X	X	X
11	UP1_DEF0_OR	This bit controls the up1 def[0] strap input of the SerDes block. [OR]	RW	0:Hard	–	X	X	X

Table 150: General-Purpose Hardware Control 0 Register (*gp_hw_ctl0*, Offset 0x8bc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
10	LED_ACT_SEL	When this bit is set, the 'S' device Rx activity feeding into the traffic LED is based on packets received from the SerDes (i.e., before the L2 filter). When this bit is '0', 5708S Rx activity feeding into the traffic LED is based on packets accepted by the EMAC (i.e., after the L2 filter). This bit must be set to '0' for the copper device.	RW	0	-	B 1 +	-	-
10–8	RESERVED	[OR]	RW	0	-	-	X X	
9	GRC_BNK_FREE_FIX	When this bit is set, a fix to the GRC state machine bank free indicator is enabled to ensure that a free bank can be re-assigned in all cases when the target transaction is complete. When this bit is '0', a situation can arise such that if one of the two banks is stuck accessing a target that does not complete, the second one will not be re-used and a GRC lock-up can occur.	RW	0	-	B 1 +	-	-
8	ENA_SEL_VAUX_B_IN_L2	When this bit is set, the SEL_VAUX_B output will assert when the PCIE is in L2, but before PERST_B has been asserted. When this bit is '0', SEL_VAUX_B will only assert once PERST_B has asserted while the PCIE is in L2/L3 mode. This prevents VAUX overcurrent while the chip is in L2, waiting for PERST_B to assert and the following main power down. This bit is used to enable the fix to gate SIGDET with the LINKRDY input.	RW	0	-	B 0 +	-	-
7	ENA_CORE_RST_OR	When this bit is set a core reset will be generated when ON_MAIN_PWR_PERST_B pin goes low in addition to when it is de-asserted (goes GOING AWAY high), but only when the chip is not prepared for power down. Prepared is the condition where SPIO2 "pin" is driving low and the PCIE is in L2/L3 power state. [OR]	RW	0:Hard	-	X X X		
6	HC_CNTL_TMOUT_CTR_RST	When this bit is set, an optimization is enabled in the HC state-machine to reduce, but not eliminate, the occurrence of redundant interrupts due to late-arriving hardware status updates as the status record is being formed.	RW	0	-	B 1 +	-	-
6–0	RESERVED		RO	0	-	-	X X	
5	RESERVED		RO	0	-	X	-	-

Table 150: General-Purpose Hardware Control 0 Register (*gp_hw_ctl0*, Offset 0x8bc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
4	FLASH_SAMP_SCLK_NE GEDGE	When this bit is set, an optimization is enabled in the Flash interface state machine to sample read data on the negative edge of SCLK. Because external flash devices output data on the negative edge of SCK, this allows for a full cycle round-trip time from launching of SCK falling edge to the sampling of data on the next falling edge. When this bit is '0' read data is sampled on the rising edge of SCLK and the maximum recommended Flash clock frequency is 16 MHz.	RW	0	-	B 1 +	-	-
3	PLL_BYPASS	When this bit is set, PLL Bypass of PCI mode is enabled.	RW	0	A 2 +	-	-	-
3	RVMII_MODE	When this bit is set and RMII_MODE is clear, the UMP port will be put into "PHY" clocking/timing mode where it will emulate the timing of a PHY MII connection. The clock inputs become clock outputs and the data pins of the MII are synchronous to those outputs. In this mode, only single UMP can be connected to a BMC. After a change to this bit, the UMP interface should be reset.	RW	0	-	X	-	-
2	RMII_CRSDV_SEL	When this bit is set, the UMP RMII interface will choose UMP_CRS input for the CRS_DV RMII input signal. When this bit is '0', the UMP RMII interface will choose UMP_RXDV input for the CRS_DV RMII input signal.	RW	0	-	B 0 +	-	-
2	PLL_BYPASS_133	When this bit is set, PLL Bypass of PCI33 mode is enabled.	RW	0	A 2 +	-	-	-
1	RMII_MODE	When this bit is set, the UMP interface runs in RMII mode, otherwise it runs in MII mode.	RW	0	X	X	-	-
0	TX_DRIVE	When this bit is set, the UMP interface TX pins are enabled to drive, otherwise the UMP interface TX pins drive control is based on the automatic drive enable (if enabled) in the UMP block.	RW	0	X	X	-	-

GENERAL-PURPOSE HARDWARE CONTROL 1 REGISTER (GP_HW_CTL1, OFFSET 0x8c0)***Table 151: General-Purpose Hardware Control 1 Register (gp_hw_ctl1, Offset 0x8c0)***

Bit	Name	Description	Mode	Reset	06	08	09	16
31–4	RESERVED		RO	0	X	X	–	–
31–0	RESERVED		RO	0	–	–	X	X
3	0_PCIE_LOOPBACK	When this bit is set, the PCIE block goes into loopback mode so that each TX channel loops back to its RX channel. At the same time, bits [39–36] of the PCI address are forced to 0xf. When this bit is set, then master cycles are wrapped back to the modified address and executed by the chip as a target cycle.	RW	0	X	X	–	–
2	1_PWR_IND_PRSNT	When this bit is set, the PCIE block indicates in its configuration space that the Power indicator is present and implemented. If this bit is clear, the PCIE block does not indicate that the Power indicator is implemented.	RW	0	X	X	–	–
1	1_ATTN_IND_PRSNT	When this bit is set, the PCIE block indicates in its configuration space that the Attention indicator is present and implemented. If this bit is clear, the PCIE block does not indicate that the Attention indicator is implemented.	RW	0	X	X	–	–
0	1_ATTN_BTN_PRSNT	When this bit is set, the PCIE block indicates in its configuration space that the Attention button is present and implemented. If this bit is clear, the PCIE block does not indicate that the Attention button is implemented.	RW	0	X	X	–	–

MISC NEW CORE CONTROL REGISTER (MISC_NEW_CORE_CTL, OFFSET 0x8c8)*Table 152: MISC New Core Control Register (misc_new_core_ctl, offset 0x8c8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–17	RESERVED_TC	This register is for general purpose hardware control for current elements in Xinan. The registers are reset by toe core reset.	RW	0	–	–	X	X
16	DMA_ENABLE	When this bit is set DMA activity will be enabled on the port.	RW	0	–	–	A	X
					1			+
15–2	RESERVED_CMN	This register is for general purpose hardware control for current elements in Xinan. The registers are reset by common core reset. [SHARE]	RW	0:CMN	–	–	X	X
1	LINK_HOLDOFF_REQ	This bit is written to a '1' to request that the PCIE link not begin training yet. Software should set this bit, and then check the LINK HOLDOFF SUCCESS bit. If LINK HOLDOFF SUCCESS is set, configure the PCIE link and then clear this bit. If LINK HOLDOFF SUCCESS is not set, the PCIE link has started training, therefore, no further configuration is possible. [SHARE]	RW	0:CMN	–	–	X	X
0	LINK_HOLDOFF_SUCCESS	This bit indicates the PCIE link is successfully being held from starting training. This is used in conjunction with LINK HOLDOFF REQ. [SHARED]	RO	0:CMN	–	–	X	X

MISC CHECKSUM 16 ERROR REGISTER (MISC_CS16_ERR, OFFSET 0x8E0)

This register controls and provides status on the cs16 data integrity feature.

Table 153: MISC Checksum 16 Error Register (misc_cs16_err, offset 0x8e0)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–25	RESERVED		RO	0	–	–	X	X
24	STA_CP	When this bit is '1', it indicates a cs16 failure in the CP block.	WC	0	–	–	X	X
23	STA_COM	When this bit is '1', it indicates a cs16 failure in the COM block.	WC	0	–	–	X	X
22	STA_RBDC	When this bit is '1', it indicates a cs16 failure in the RBDC block.	WC	0	–	–	X	X
21	STA_TBDR	When this bit is '1', it indicates a cs16 failure in the TBDR block.	WC	0	–	–	X	X
20	STA_CTX	When this bit is '1', it indicates a cs16 failure in the CTX block.	WC	0	–	–	X	X
19	STA_EMAC	When this bit is '1', it indicates a cs16 failure in the EMAC block.	WC	0	–	–	X	X
18	STA_TDMA	When this bit is '1', it indicates a cs16 failure in the TDMA block.	WC	0	–	–	X	X
17	STA_RDMA	When this bit is '1', it indicates a cs16 failure in the RDMA block.	WC	0	–	–	X	X
16	STA_PCI	When this bit is '1', it indicates a cs16 failure in the PCI block. [SHARE]	WC	0:CMN	–	–	X	X
15–9	RESERVED		RO	0	–	–	X	X
8	ENA_CP	When this bit is '1', it enables cs16 checking in the CP block.	RW	0	–	–	X	X
7	ENA_COM	When this bit is '1', it enables cs16 checking in the COM block.	RW	0	–	–	X	X
6	ENA_RBDC	When this bit is '1', it enables cs16 checking in the RBDC block.	RW	0	–	–	X	X
5	ENA_TBDR	When this bit is '1', it enables cs16 checking in the TBDR block.	RW	0	–	–	X	X
4	ENA_CTX	When this bit is '1', it enables cs16 checking in the CTX block.	RW	0	–	–	X	X
3	ENA_EMAC	When this bit is '1', it enables cs16 checking in the EMAC block.	RW	0	–	–	X	X
2	ENA_TDMA	When this bit is '1', it enables cs16 checking in the TDMA block.	RW	0	–	–	X	X
1	ENA_RDMA	When this bit is '1', it enables cs16 checking in the RDMA block.	RW	0	–	–	X	X
0	ENA_PCI	When this bit is '1', it enables cs16 checking in the PCI block. [SHARE]	RW	0:CMN	–	–	X	X

MISC DUAL MEDIA CONTROL REGISTER (MISC_DUAL_MEDIA_CTRL, OFFSET 0x8EC)*Table 154: MISC Dual Media Control Register (misc_dual_media_ctrl, offset 0x8ec)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	RESERVED		RO	0	X	X	-	-
31-30	RESERVED		RO	0	-	-	X	X
29-26	PHY_SERDES_IDDQ	This field is used to completely power down any of the PHYs including the signal detect logic. It should be used in applications where PHY configuration is known, it will not change and minimal possible power consumption is desired.	RW	0:CMN	-	-	X	X
		Value Name Description						
		1 SER1_IDDQ This bit sets the IDDQ on SERDES 1.						
		2 SER0_IDDQ This bit sets the IDDQ on SERDES 0.						
		4 PHY1_IDDQ This bit sets the IDDQ on PHY1.						
		8 PHY0_IDDQ This bit sets the IDDQ on PHY0.						
25	STRAP_OVERRIDE	When this bit is set PORT SWAP and PHY CTRL[2:0] bits of this register are used to control PHYs instead of the corresponding straps. [SHARE]	RW	0:CMN	-	-	X	X
24	PORT_SWAP	This bit acts as a port swap control for "C" and "S" devices. For "X" devices this bit should be understood as an extension of the PHY CTRL[2:0] field and used together with it to specify which MAC connects to which PHY and as well the active PHYs. [SHARE]	RW	0:CMN	-	-	X	X
23-21	PHY_CTRL	This field together with PORT SWAP bit selects active PHYs and specifies which MAC is connected to which PHY. [SHARE]	RW	0:CMN	-	-	X	X
20	PHY0_RST	When set this bit resets PHY0. Should be held high by software for at least TBD ns. Should be asserted before PHY is put in power down mode and de-asserted at least TBD ms after PHY was taken out of power down mode. This bit has effect only on "X" or "C" devices. [SHARE]	RW	0:CMN	-	-	X	X
19	PHY1_RST	When set this bit resets PHY1. Should be held high by software for at least TBD ns. Should be asserted before PHY is put in power down mode and de-asserted at least TBD ms after PHY was taken out of power down mode. This bit has effect only on "X" or "C" devices. [SHARE]	RW	0:CMN	-	-	X	X

Table 154: MISC Dual Media Control Register (misc_dual_media_ctrl, offset 0x8ec) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
18	SERDES0_RST	When set this bit resets SERDES0. Should be held high by software for at least 200 ns. Should be asserted before SerDes is put in power down mode and de-asserted at least 1 ms after SerDes was taken out of power down mode. If the LCPLL was powered down SerDes reset should be de-asserted at least 20 ms after LCPLL RST is de-asserted. This bit has effect only on "X" or "S" devices. [SHARE]	RW	0:CMN	-	-	X	X
17	SERDES1_RST	When set this bit resets SERDES1. Should be held high by software for at least 200 ns. Should be asserted before SERDES is put in power down mode and de-asserted at least 1 ms after SerDes was taken out of power down mode. If the LCPLL was powered down SERDES reset should be de-asserted at least 20ms after LCPLL RST is de-asserted. This bit has effect only on "X" or "S" devices. [SHARE]	RW	0:CMN	-	-	X	X
16	LCPLL_RST	When set this bit resets LCPLL. Should be held high by software for at least 30 μ s. Should be asserted before LCPLL is put in power down mode and de-asserted after LCPLL was taken out of power down mode. This bit has effect only on "X" or "S" devices. [SHARE]	RW	0:CMN	-	-	X	X
15	PHY0_SIGDET	Valid even when PHY is powered down. Used for the link activity detection. This bit is valid only for "X" or "C" devices. [SHARE]	WC	0:CMN	-	-	X	X
14	PHY1_SIGDET	Valid even when PHY is powered down. Used for the link activity detection. This bit is valid only for "X" or "C" devices. [SHARE]	WC	0:CMN	-	-	X	X
13	SERDES0_SIGDET	Valid even when SerDes is powered down. Used for the link activity detection. This bit is valid only for "X" or "S" device. [SHARE]	WC	0:CMN	-	-	X	X
12	SERDES1_SIGDET	Valid even when SerDes is powered down. Used for the link activity detection. This bit is valid only for "X" or "S" devices. [SHARE]	WC	0:CMN	-	-	X	X
11	PORT_SWAP_PIN	This bit reflects the value of PORT SWAP PIN. [SHARE]	RO	0:CMN	-	-	X	X
10-8	PHY_CTRL_STRAP	These bits reflects the value of PHY CTRL STRAP[2:0]. [SHARE]	RO	0:CMN	-	-	X	X

Table 154: MISC Dual Media Control Register (misc_dual_media_ctrl, offset 0x8ec) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7–0	BOND_ID	These bits are connected to the OTP ROM MEDIA CONTROL[7:0] outputs. BOND ID[7:4] are reserved for the future use. BOND ID[3:0] indicate type of the device. These bits have same value as the BOND ID bits in ID register (0x808) and they are repeated here for the software convenience. [SHARE]	RO	0:CMN	–	–	X	X
Value Name Description								
0	X	All 4 PHYs are available.						
3	C	Two copper PHYs are available.						
12	S	Two SerDes PHYS are available.						

MISC BIST CS2 REGISTER (MISC_BIST_CS2, OFFSET 0x91c)

Table 155: MISC BIST CS2 Register (misc_bist_cs2, offset 0x91c)

Bit	Name	Description	Mode	Reset	06	08	09	16															
31–10	RESERVED		RO	0	X	X	X	X															
9–0	RESRVED		RO	0	X	X	—	—															
9	MBIST_GO	BIST has passed when set to 1 and MBIST DONE is set.	RO	0	—	—	X	X															
8	MBIST_DONE	BIST has completed when set to 1.	RO	0	—	—	X	X															
7–4	RESERVED		RO	0	—	—	X	X															
3	MBIST_ASYNC_RESET	Asynchronous reset for BIST controller. Drive high before enabling BIST.	RW	0	—	—	X	X															
2–1	BIST_SETUP	Controls BIST options.	RW	0x2	—	—	X	X															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>SHORT RUN</td><td>This mode is not supported in Xinan.</td></tr> <tr> <td>1</td><td>LONG RUN</td><td>This mode is not supported in Xinan.</td></tr> <tr> <td>2</td><td>DEFAUL T RUN</td><td>This mode is supported in Xinan.</td></tr> <tr> <td>3</td><td>NORM AL RUN</td><td>This mode is not supported in Xinan.</td></tr> </tbody> </table>									Value	Name	Description	0	SHORT RUN	This mode is not supported in Xinan.	1	LONG RUN	This mode is not supported in Xinan.	2	DEFAUL T RUN	This mode is supported in Xinan.	3	NORM AL RUN	This mode is not supported in Xinan.
Value	Name	Description																					
0	SHORT RUN	This mode is not supported in Xinan.																					
1	LONG RUN	This mode is not supported in Xinan.																					
2	DEFAUL T RUN	This mode is supported in Xinan.																					
3	NORM AL RUN	This mode is not supported in Xinan.																					
0	MBIST EN	BIST is run when this bit is set to 1	RW	0	—	—	X	X															

MISC PARITY ERROR STATUS REGISTER (MISC_PERR_STATUS0, OFFSET 0x944)*Table 156: MISC Parity Error Status Register (misc_perr_status0, offset 0x944)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	TSCH_LR_PERR	Parity Status	RO	0	–	–	X	X
30	TBDC_PERR	Parity Status	RO	0	–	–	X	X
29	TDMA_PERR	Parity Status	RO	0	–	–	X	X
28	THBUF_PERR	Parity Status	RO	0	–	–	X	X
27	TPBUF_PERR	Parity Status	RO	0	–	–	X	X
26	RV2P_CB0REGS_PERR	Parity Status	RO	0	–	–	X	X
25	RV2P_CB1REGS_PERR	Parity Status	RO	0	–	–	X	X
24	RV2P_P1IRAM_PERR	Parity Status	RO	0	–	–	X	X
23	RV2P_P2IRAM_PERR	Parity Status	RO	0	–	–	X	X
22	RBUF_DATAMEM_PERR	Parity Status	RO	0	–	–	X	X
21	RBUF_PTRMEM_PERR	Parity Status	RO	0	–	–	X	X
20	RPC_DFIFOMEM_PERR	Parity Status	RO	0	–	–	X	X
19	RPM_DFIFOMEM_PERR	Parity Status	RO	0	–	–	X	X
18	MQ_CTX_PERR	Parity Status	RO	0	–	–	X	X
17	CS_TMEM_PERR	Parity Status	RO	0	–	–	X	X
16	TXP_SCPAD_PERR	Parity Status	RO	0	–	–	X	X
15	TXP_CTXC_PERR	Parity Status	RO	0	–	–	X	X
14	TPAT_SCPAD_PERR	Parity Status	RO	0	–	–	X	X
13	RXP_SCPAD_PERR	Parity Status	RO	0	–	–	X	X
12	RXP_CTXC_PERR	Parity Status	RO	0	–	–	X	X
11	RXP_RBUFC_PERR	Parity Status	RO	0	–	–	X	X
10	CP_SCPAD_PERR	Parity Status	RO	0	–	–	X	X
9	CP_CTXC_PERR	Parity Status	RO	0	–	–	X	X
8	COM_SCPAD_PERR	Parity Status	RO	0	–	–	X	X
7	COM_CTXC_PERR	Parity Status	RO	0	–	–	X	X
6	CTX_MIRROR_PERR	Parity Status	RO	0	–	–	X	X
5	CTX_CACHE_PERR	Parity Status	RO	0	–	–	X	X
4	CTX_PGTBL_PERR	Parity Status	RO	0	–	–	X	X
3	CTX_USAGE_CNT_PERR	Parity Status	RO	0	–	–	X	X
2	RPM_ACPIBEMEM_PERR	Parity Status	RO	0	–	–	X	X
1	CP_DMAE_PERR	Parity Status	RO	0	–	–	X	X
0	COM_DMAE_PERR	Parity Status	RO	0	–	–	X	X

MISC PARITY ERROR STATUS 1 REGISTER (MISC_PERR_STATUS1, OFFSET 0x948)**Table 157: MISC Parity Error Status 1 Register (misc_perr_status1, offset 0x948)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–30	RESERVED		RO	0	X	X	X	X
29–0	RESERVED		RO	0	X	X	–	–
29	MQ_IDX_PERR	Parity Status	RO	0	–	–	X	X
28	RV2PCSQ_PERR	Parity Status	RW	0	–	–	X	X
27	RV2PCS_TMEM_PERR	Parity Status	RW	0	–	–	X	X
26	RLUP_CID_PERR	Parity Status	RO	0	–	–	X	X
25	CSQ_PERR	Parity Status	RO	0	–	–	X	X
24	CPQ_PERR	Parity Status	RO	0	–	–	X	X
23	RV2PMQ_PERR	Parity Status	RO	0	–	–	X	X
22	RV2PTQ_PERR	Parity Status	RO	0	–	–	X	X
21	RXPCQ_PERR	Parity Status	RO	0	–	–	X	X
20	COMXQ_PERR	Parity Status	RO	0	–	–	X	X
19	TSCHQ_PERR	Parity Status	RO	0	–	–	X	X
18	TBDRQ_PERR	Parity Status	RO	0	–	–	X	X
17	TASQ_PERR	Parity Status	RO	0	–	–	X	X
16	RDMAQ_PERR	Parity Status	RO	0	–	–	X	X
15	RV2PPQ_PERR	Parity Status	RO	0	–	–	X	X
14	RXPQ_PERR	Parity Status	RO	0	–	–	X	X
13	RLUPQ_PERR	Parity Status	RO	0	–	–	X	X
12	COMQ_PERR	Parity Status	RO	0	–	–	X	X
11	COMTQ_PERR	Parity Status	RO	0	–	–	X	X
10	TXPQ_PERR	Parity Status	RO	0	–	–	X	X
9	TDMAQ_PERR	Parity Status	RO	0	–	–	X	X
8	MCPQ_PERR	Parity Status	RO	0	–	–	X	X
7	TPATQ_PERR	Parity Status	RO	0	–	–	X	X
6	HC_CONSUMSTB_PERR	Parity Status	RO	0	–	–	X	X
5	HC_PRODUCSTB_PERR	Parity Status	RO	0	–	–	X	X
4	HC_MSIX_PERR	Parity Status	RO	0	–	–	X	X
3	HC_STATS_PERR	Parity Status	RO	0	–	–	X	X
2	RDMA_DFIFO_PERR	Parity Status	RO	0	–	–	X	X
1	RESERVED		RO	0	–	–	X	X
0	RBDC PERR	Parity Status	RO	0	–	–	X	X

MISC PARITY ERROR STATUS 2 REGISTER (MISC_PERR_STATUS2, OFFSET 0x94c)*Table 158: MISC Parity Error Status 2 Register (misc_perr_status2, offset 0x94c)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–7	RESERVED		RO	0	X	X	X	X
6–0	RESERVED		RO	0	X	X	–	–
6	PCIE_REPLY_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X
5	HB_MEM_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X
4	MCP_SCPAD_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X
3	MCP_ROM_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X
2	UMP_RX_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X
1	UMP_TX_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X
0	TGT_FIFO_PERR	Parity Status [OR]	RO	0:CMN	–	–	X	X

DMA ENGINE (DMA) REGISTERS

DMA COMMAND REGISTER (DMA_COMMAND, OFFSET 0xC00)

Table 159: DMA Command Register (dma_command, Offset 0xc00)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–1	RESERVED		RO	0	X	X	X	X
0	ENABLE	This bit indicates the enable state of the DMA Engine and is controlled by the DMA_ENGINE_ENABLE bit of the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232.	RO	0	X	X	X	X

DMA STATUS REGISTER (DMA_STATUS, OFFSET 0xc04)*Table 160: DMA Status Register (dma_status, Offset 0xc04)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	—	—
31–5	RESERVED		RO	0	—	—	X	X
25	BIG_WRITE_RETRY_AFTER_DATA_STAT	This is a generic statistic output for the count of write transfers on maximum buffer sizes that retried after data had been transferred. This can be used to determine the amount of buffering the bridge is doing. This output toggles once each time the event happens.	RO	0	X	X	—	—
24	BIG_WRITE_DELAY_PCI_CLKS_STAT	This is a generic statistic output for the count of PCI clocks each write requestor spends with its request active, but no data transfer occurring for transfers of the maximum size. This output toggles once for every 64 PCI clocks that every requestor has waited.	RO	0	X	X	—	—
23	BIG_WRITE_TRANSFERS_STAT	This is a generic statistic output for the count of PCI write transfers that were of the maximum size. Some of the statistics below are only gathered on these types of transfers. This output toggles once for each event.	RO	0	X	X	—	—
22	WRITE_DELAY_PCI_CLKS_STAT	This is a generic statistic output for the count of PCI clocks each write requestor spends with its request active, but no data transfer occurring. This output toggles once for every 64 PCI clocks that each requestor has waited.	RO	0	X	X	—	—
21	WRITE_TRANSFERS_STAT	This is a generic statistic output for the count total of PCI write transfers that have been performed. This output toggles once for each event.	RO	0	X	X	—	—
20	BIG_READ_RETRY_AFTER_DATA_STAT	This is a generic statistic output for the count of read transfers on maximum buffer sizes that retried after data had been transferred. This can be used to determine the amount of read-ahead the bridge is doing. This output toggles once for each event.	RO	0	X	X	—	—

Table 160: DMA Status Register (dma_status, Offset 0xc04) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
19	BIG_READ_DELAY_PCI_CLKS_STAT	This is a generic statistic output for the count of PCI clocks each read requestor spends with its request active, but no data transfer occurring for transfers of the maximum buffer size. This output toggles once for every 64 PCI clocks that each requestor has waited.	RO	0	X	X	-	-
18	BIG_READ_TRANSFERS_STAT	This is a generic statistic output for the count of PCI read transfers that were of the max buffer size. Some other statistics are only accumulated for these types of transfers. This output toggles once for each big transfer occurs.	RO	0	X	X	-	-
17	READ_DELAY_PCI_CLKS_STAT	This is a generic statistic output for the count of PCI clocks each read requestor spends with its request active, but no data transfer occurring. This output toggles once each for every 64 PCI clocks that each requestor has waited.	RO	0	X	X	-	-
16	READ_TRANSFERS_STAT	This is a generic statistic output for the total count of PCI read transfers that have been performed. This output toggles once each time a new DMA transfer has occurred.	RO	0	X	X	-	-
15–1	RESERVED		RO	0	X	X	-	-
4	BME	This bit indicates if the bus master enable was set for a port.	RO	0	-	-	X	X
3–1	RESERVED		RO	0	-	-	X	X
0	PAR_ERROR_STATE	When this bit is 1, it indicates that the DMA engine has detected that the parity error attention condition exists and is limiting DMA operations to the HC block only.	RO	0	X	X	X	X

DMA CONFIGURATION REGISTER (DMA_CONFIG, OFFSET 0xc08)**Table 161: DMA Configuration Register (dma_config, Offset 0xc08)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	—	—
31	NO_64SWAP_EN	Setting this bit disables 64bit swapping on the 128bit bus coming into the DMA engine. [SHARE]	RW	0:CMN	—	—	X	X
30–20	RESERVED		RO	0	—	—	X	X
27–24	BIG_SIZE	This bit mask is used to determine what a BIG DMA size is. The same value is used for write and read. If the size ANDed with this mask is high, then the transfer is BIG.	RW	0x8	X	X	—	—
Value Name								
	0x0	None	—					
	0x1	64	—					
	0x2	128	—					
	0x4	256	—					
	0x8	512	—					
23	PCI_FAST_CLK_CMP	This bit should be set for proper DMA operation above 100 MHz in PCI-X mode and clear for speeds below 66 MHz. When this bit is set, the pci_dr_comp code is delayed by one clock to provide adequate hold time for this data as completions are synchronized to the core clock domain.	RW	0x1	X	X	—	—
22–20	PCI_CLK_CMP_BITS	The PCLK block generates a 4-bit vector, depending on the PCI_CLK frequency. Set this field to match the lowest value of that 4-bit vector for which clock compensation that is to be enabled. Note: Only values up to 0x7 are supported by this field. The 3-bit value is 0 extended for comparison purposes. The CNTL_PCI_COMP_DLY bit must be set for this to be enabled. The normal value for this register should be 0x2, which compensates for frequencies of about 40 MHz and below.	RW	0	X	X	—	—
19–16	NO_WCHANS_IN_USE	This nibble sets the number of DMA write channels that are available for use by the write clients of the DMAE.	RW	0x1	X	X	—	—
19	MAX_RSS_EN	Setting this bit allows the DME engine to override the maximum read request setting of the system and use the setting in MAX_RSS. [SHARE]	RW	0:CMN	—	—	X	X

Table 161: DMA Configuration Register (dma_config, Offset 0xc08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
18–16	MAX_RSS	This field sets the maximum read request for a DMA read operation and is enabled by setting the MAX_RSS_EN bit. [SHARE]	RW	0:CMN	—	—	X	X
		Value Name	Description					
		0	128B	—				
		1	256B	—				
		2	512B	—				
		3	1024B	—				
		4	2048B	—				
		5	4096B	—				
15–12	NO_RCHANS_IN_USE	This nibble sets the number of DMA read channels that are available for use by the read clients of the DMAE.	RW	0x1	X	X	—	—
15	MAX_PL_EN	Setting this bit allows the DMA engine to override the maximum payload setting of the system and use the value in MAX_PL. [SHARE]	RW	0:CMN	—	—	X	X
14–12	MAX_PL	This field sets the maximum payload for a DMA write operation and is enabled by setting the MAX_PL_EN bit. [SHARE]	RW	0:CMN	—	—	X	X
		Value Name	Description					
		0	128B	—				
		1	256B	—				
		2	512B	—				
11	CNTL_PCI_COMP_DLY	This bit is set to enable clock compensation for operation at slow PCI clock speeds, along with the PCI_CLK_CMP_BITS field. If this bit is clear, then no slow speed compensation is done and the PCI_CLK_CMP_BITS field is ignored.	RW	0	X	X	—	—
11–6	RESERVED		RO	0	—	—	X	X
10	CNTL_PING_PONG_DMA	This bit enforces Ping-Pong mode in the DMA arbiter. In this mode, the arbiter will lock into one read and one write DMA channel and execute each until complete, then select the next operation. This mode limits the bridge to two DMA addresses from the NetXtreme II at any one time. The mode approximates normal BCM5700 operations.	RW	0	X	X	—	—
9	RESERVED		RO	0	X	X	—	—
8	CNTL_FPGA_MODE	Setting this bit allows the PCI clock to be up to 50 times faster than the CORE clock. This is used during FPGA emulation. Without this bit set, the PCI clock can be no more than twice CORE clock.	RW	0	X	X	—	—

Table 161: DMA Configuration Register (dma_config, Offset 0xc08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7	CNTL_TWO_DMA	This bit enforces Two-DMA-At-A-Time mode in the DMA arbiter, which is quite different from ONE_DMA mode. In this mode, 8 priority bins are separated into the read and write sets, and the arbiter will alternate between one from the read set and one from the write set. This mode does not limit the number of DMA addresses that the bridge will see from the NetXtreme II at any one time.	RW	0	X	X	-	-
6	ONE_DMA	This bit enforces One-DMA-At-A-Time mode in the DMA arbiter. This is used as a debug tool. When this bit is set, the DMA will pick a DMA operation based on fixed priority and then process that DMA operation until it is satisfied. This means that the bridge will see only one DMA address from the NetXtreme II at any one time.	RW	0	X	X	-	-
5	CNTL_WORD_SWAP	Setting this bit enables word swap for data words transferred by users indicating control swapping.	RW	0	X	X	X	X
4	CNTL_BYTE_SWAP	Setting this bit enables byte swap for data words transferred by users indicating control swapping.	RW	0	X	X	X	X
3-2	RESERVED		RO	0	X	X	X	X
1	DATA_WORD_SWAP	Setting this bit enables word swap for data words transferred by users indicating data swapping.	RW	0	X	X	X	X
0	DATA_BYTE_SWAP	Setting this bit enables byte swap for data words transferred by users indicating data swapping.	RW	0	X	X	X	X

DMA BLACKOUT REGISTER (DMA_BLACKOUT, OFFSET 0xc0c)**Table 162: DMA Blackout Register (dma_blackout, Offset 0xc0c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31-24	RESERVED		RO	0	X	X	X	X
23-16	WR_RETRY_BLACKOUT	When this field is programmed to a value other than 0, Write-Retry-Blackout mode will be enabled. This means that each write DMA channel will deassert its request for the number of PCI clocks specified in this register on the first retry of a transfer after data has been transferred. Each DMA channel will maintain its own count, but all will refer to this count to determine when to re-assert its request. The intent is to allow the bridge enough time to empty its buffers to accept more data before sending more data.	RW	0	X	X	-	-
23-0	RESERVED		RO	0	-	-	X	X



Table 162: DMA Blackout Register (dma_blackout, Offset 0xc0c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–8	2ND_RD_RETRY_BLACKOUT	This field is used when RD_RETRY_BLACKOUT is not 0. This value will be used as the blackout count for all retries after data has transferred. This value has no effect on split completions.	RW	0	X	X	–	–
7–0	RD_RETRY_BLACKOUT	When this field is programmed to a value other than 0, Read-Retry-Blackout mode will be enabled. This means that each read DMA channel will deassert its request for the number of PCI clocks specified in this register on the first retry of a transfer. Each DMA channel will maintain its own count, but all will refer to this count to determine when to re-assert its request. The intent is to give the bridge time to get the data needed to fulfill the read without retrying needlessly. This value has no effect on split completions.	RW	0	X	X	–	–

DMA READ MASTER SETTING 0 REGISTER (DMA_READ_MASTER_SETTING_0, OFFSET 0xC10)**Table 163: DMA Read Master Setting 0 Register (dma_read_master_setting_0, offset 0xc10)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	CTX_PARAM_EN	This bit enables the corresponding parameters to be used for CTX.	RW	0	–	–	X	X
30–28	CTX_TRAFFIC_CLASS	This field sets the traffic class for CTX.	RW	0	–	–	X	X
27	RESERVED		RO	0	–	–	X	X
26	CTX_PRIORITY	This bit sets the priority of CTX for arbitration purposed.	RW	0	–	–	X	X
25	CTX_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for CTX.	RW	0	–	–	X	X
24	CTX_NO_SNOOP	This bit sets the No Snoop parameter for CTX.	RW	0	–	–	X	X
23	TDMA_PARAM_EN	This bit enables the corresponding parameters to be used for TDMA.	RW	0	–	–	X	X
22–20	TDMA_TRAFFIC_CLASS	This field sets the traffic class for TDMA.	RW	0	–	–	X	X
19	RESERVED		RO	0	–	–	X	X
18	TDMA_PRIORITY	This bit sets the priority of TDMA for arbitration purposed.	RW	0	–	–	X	X
17	TDMA_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for TDMA.	RW	0	–	–	X	X
16	TDMA_NO_SNOOP	This bit sets the No Snoop parameter for TDMA.	RW	0	–	–	X	X
15	RBDC_PARAM_EN	This bit enables the corresponding parameters to be used for RBDC.	RW	0	–	–	X	X
14–12	RBDC_TRAFFIC_CLASS	This field sets the traffic class for RBDC.	RW	0	–	–	X	X
11	RESERVED		RO	0	–	–	X	X
10	RBDC_PRIORITY	This bit sets the priority of RBDC for arbitration purposed.	RW	0	–	–	X	X
9	RBDC_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for RBDC.	RW	0	–	–	X	X
8	RBDC_NO_SNOOP	This bit sets the No Snoop parameter for RBDC.	RW	0	–	–	X	X
7	TBDC_PARAM_EN	This bit enables the corresponding parameters to be used for TBDC.	RW	0	–	–	X	X
6–4	TBDC_TRAFFIC_CLASS	This field sets the traffic class for TBDC.	RW	0	–	–	X	X
3	RESERVED		RO	0	–	–	X	X
2	TBDC_PRIORITY	This bit sets the priority of TBDC for arbitration purposed.	RW	0	–	–	X	X
1	TBDC_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for TBDC.	RW	0	–	–	X	X
0	TBDC_NO_SNOOP	This bit sets the No Snoop parameter for TBDC.	RW	0	–	–	X	X

DMA READ MASTER SETTING 1 REGISTER (DMA_READ_MASTER_SETTING_1, 0xc14)**Table 164: DMA Read Master Setting 1 Register (dma_read_master_setting_1, 0xc14)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–0	RESERVED		RO	0	X	X	–	–
15	CP_PARAM_EN	This bit enables the corresponding parameters to be used for CP.	RW	0	–	–	X	X
14–12	CP_TRAFFIC_CLASS	This field sets the traffic class for CP.	RW	0	–	–	X	X
11	RESERVED		RO	0	–	–	X	X
10	CP_PRIORITY	This bit sets the priority of CP for arbitration purposes.	RW	0	–	–	X	X
9	CP_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for CP.	RW	0	–	–	X	X
8	CP_NO_SNOOP	This bit sets the No Snoop parameter for CP.	RW	0	–	–	X	X
7	COM_PARAM_EN	This bit enables the corresponding parameters to be used for COM.	RW	0	–	–	X	X
6–4	COM_TRAFFIC_CLASS	This bit sets the traffic class for COM.	RW	0	–	–	X	X
3	RESERVED		RO	0	–	–	X	X
2	COM_PRIORITY	This bit sets the priority of COM for arbitration purposes.	RW	0	–	–	X	X
1	COM_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for COM.	RW	0	–	–	X	X
0	COM_NO_SNOOP	This bit sets the No Snoop parameter for COM.	RW	0	–	–	X	X

DMA WRITE MASTER SETTING 0 REGISTER (DMA_WRITE_MASTER_SETTING_0, OFFSET 0xc18)

Table 165: DMA Write Master Setting 0 Register (dma_write_master_setting_0, offset 0xc18)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	CTX_PARAM_EN	This bit enables the corresponding parameters to be used for CTX.	RW	0	–	–	X	X
30–28	CTX_TRAFFIC_CLASS	This field sets the traffic class for CTX.	RW	0	–	–	X	X
27	CTX_CS_VLD	This bit indicates whether the data integrity checksum from the CTX block is valid or not. SPLIT: This bit is sent to the PCIE block when CTX is the master on the bus.	RW	0x1	–	–	X	X
26	CTX_PRIORITY	This bit sets the priority of CTX for arbitration purposed.	RW	0	–	–	X	X
25	CTX_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for CTX.	RW	0	–	–	X	X
24	CTX_NO_SNOOP	This bit sets the No Snoop parameter for CTX.	RW	0	–	–	X	X
23–16	RESERVED		RO	0	–	–	X	X
15	RDMA_PARAM_EN	This bit enables the corresponding parameters to be used for RDMA.	RW	0	–	–	X	X
14–12	RDMA_TRAFFIC_CLASS	This field sets the traffic class for RDMA.	RW	0	–	–	X	X
11	RDMA_CS_VLD	This bit indicates whether the data integrity checksum from the RDMA block is valid or not. SPLIT: This bit is sent to the PCIE block when RDMA is the master on the bus.	RW	0x1	–	–	X	X
10	RDMA_PRIORITY	This bit sets the priority of RDMA for arbitration purposed.	RW	0	–	–	X	X
9	RDMA_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for RDMA.	RW	0	–	–	X	X
8	RDMA_NO_SNOOP	This bit sets the No Snoop parameter for RDMA.	RW	0	–	–	X	X
7	HC_PARAM_EN	This bit enables the corresponding parameters to be used for HC.	RW	0	–	–	X	X
6–4	HC_TRAFFIC_CLASS	This field sets the traffic class for HC.	RW	0	–	–	X	X
3	HC_CS_VLD	This bit indicates whether the data integrity checksum from the HC block is valid or not. SPLIT: This bit is sent to the PCIE block when HC is the master on the bus.	RW	0x1	–	–	X	X
2	HC_PRIORITY	This bit sets the priority of HC for arbitration purposed.	RW	0	–	–	X	X

Table 165: DMA Write Master Setting 0 Register (dma_write_master_setting_0, offset 0xc18) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	HC_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for HC. These are debug bits and should never be used in functional mode since HC has status/statistics/MSI DMA that require different settings.	RW	0	-	-	X	X
0	HC_NO_SNOOP	This bit sets the No Snoop parameter for HC. These are debug bits and should never be used in functional mode since HC has status/statistics/MSI DMA that require different settings.	RW	0	-	-	X	X

DMA WRITE MASTER SETTING 1 REGISTER (DMA_WRITE_MASTER_SETTING_1, OFFSET 0xc1c)

Table 166: DMA Write Master Setting 1 Register (dma_write_master_setting_1, offset 0xc1c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–0	RESERVED		RO	0	X	X	-	-
15	CP_PARAM_EN	This bit enables the corresponding parameters to be used for CP.	RW	0	-	-	X	X
14–12	CP_TRAFFIC_CLASS	This bit sets the traffic class for CP.	RW	0	-	-	X	X
11	CP_CS_VLD	This bit indicates whether the data integrity checksum from the CP block is valid or not. SPLIT: This bit is sent to the PCIE block when CP is the master on the bus.	RW	0x1	-	-	X	X
10	CP_PRIORITY	This bit sets the priority of CP for arbitration purposed.	RW	0	-	-	X	X
9	CP_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for CP.	RW	0	-	-	X	X
8	CP_NO_SNOOP	This bit sets the No Snoop parameter for CP.	RW	0	-	-	X	X
7	COM_PARAM_EN	This bit enables the corresponding parameters to be used for COM.	RW	0	-	-	X	X
6–4	COM_TRAFFIC_CLAS S	This bit sets the traffic class for COM.	RW	0	-	-	X	X
3	COM_CS_VLD	This bit indicates whether the data integrity checksum from the COM block is valid or not.	RW	0x1	-	-	X	X
2	COM_PRIORITY	This bit sets the priority of COM for arbitration purposed.	RW	0	-	-	X	X
1	COM_RELAX_ORDER	This bit sets the Relaxed Ordering parameter for COM.	RW	0	-	-	X	X
0	COM_NO_SNOOP	This bit sets the No Snoop parameter for COM.	RW	0	-	-	X	X

DMA ARBITER REGISTER (DMA_ARBITER, OFFSET 0xc20)*Table 167: DMA Arbiter Register (dma_arbiter, offset 0xc20)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–0	RESERVED		RO	0	X	X	—	—
15–12	OUSTD_READ_REQ	These bits control how many outstanding read requests are allowed by the DMA Engine. The default is 12 which is the maximum number of read requests supported. [SHARE]	RW	0xc:CMN	—	—	X	X
11	RESERVED		RO	0	—	—	X	X
10	TIMER_MODE	Setting this bit will result in DMAE servicing one port at a time for the microseconds based on the maximum and minimum timeouts. This is for the Tier1 arbiter. [SHARE]	RW	0:CMN	—	—	X	X
9	RR_MODE	Setting this bit results in DMAE performing a simple round robin algorithm instead of Deficit Round Robin on the Tier1 Arbiter. [SHARE]	RW	0:CMN	—	—	X	X
8	ALT_MODE_EN	Setting this bit causes DMAE to use an alternate arbitration mode where multiple read requests can be scheduled before a write request. This is for the Tier2 arbiter. [SHARE]	RW	0:CMN	—	—	X	X
7	RESERVED		RO	0	—	—	X	X
6–5	RD_ARB_MODE	This field selects what kind of arbitration will be performed on the Tier0 read arbiter. [SHARE]	RW	0:CMN	—	—	X	X
Value Name Description								
0	STRICT	The arbiter will be a strict priority arbiter and will go in the order CTX->RBDC->TBDC->COM->CP->TDMA						
1	RND RBN	The arbiter will be a simple round robin arbiter and will go in the order CTX->RBDC->TBDC->COM->CP->TDMA						
2	WGT RND RBN	The arbiter will be assign a weight of 4 to CTX and perform round robin priority arbitration and will go in the order CTX->CTX->CTX->RBDC->TBDC->COM->CP->TDMA						

Table 167: DMA Arbiter Register (dma_arbiter, offset 0xc20) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
4	WR_ARB_MODE	This bit selects what kind of arbitration will be performed on the Tier0 write arbiter. [SHARE]	RW	0x1:CMN	-	-	X	X
Value Name Description								
	0 STRI CT	The arbiter will be a strict priority arbiter and will go in the order HC->CTX->COM->CP->RDMA						
	1 RND RBN	The arbiter will be a simple round robin arbiter and will go in the order HC->CTX->COM->CP->RDMA						
3	RESERVED		RO	0	-	-	X	X
2-0	NUM_READS	This field sets how many read requests can be sent out before a write request is made. To use this feature, BIT 8 of this register must be set. Setting this to 1 will result in a ping-pong between read and write operations. [SHARE]	RW	0:CMN	-	-	X	X

DMA ARBITER TIMERS REGISTER (DMA_ARB_TIMERS, OFFSET 0xC24)*Table 168: DMA Arbiter Timers Register (dma_arb_timers, offset 0xc24)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–20	TM_MAX_TIMEOUT	This register sets the maximum time in microseconds a single port is serviced by DMAE before giving access to the other port. This value is active during timer mode. This is for the Tier1 arbiter. [SHARE]	RW	0:CMN	–	–	X	X
19–12	TM_MIN_TIMEOUT	This register sets the time in microseconds a port can be inactive during timer mode before the other port starts getting serviced. This is for the Tier1 arbiter. [SHARE]	RW	0:CMN	–	–	X	X
11–8	RESERVED		RO	0	–	–	X	X
7–0	RD_DRD_WAIT_TIME	These bits set the number of core clock cycles the Read DRD arbiter will wait for a request to show on the current active port before resetting the DC quanta and switching over to the other port. This is for the Tier1 arbiter. [SHARE]	RW	0:CMN	–	–	X	X

CONTEXT MEMORY (CTX) REGISTERS

CTX COMMAND REGISTER (CTX_COMMAND, OFFSET 0x1000)

Table 169: CTX Command Register (ctx_command, Offset 0x1000)

Bit	Name	Description	Mode	Reset	06	08	09	16																																									
31–1	RESERVED		RO	0	X	X	–	–																																									
31–20	RESERVED		RO	0	–	–	X	X																																									
19–16	PAGE_SIZE	This field sets the page size for each context page allotted in the host memory.	RW	0x4	–	–	X	X																																									
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>256</td><td>256-byte page</td></tr> <tr><td>1</td><td>512</td><td>512-byte page</td></tr> <tr><td>2</td><td>1K</td><td>1-KB page</td></tr> <tr><td>3</td><td>2K</td><td>2-KB page</td></tr> <tr><td>4</td><td>4K</td><td>4-KB page</td></tr> <tr><td>5</td><td>8K</td><td>8-KB page</td></tr> <tr><td>6</td><td>16K</td><td>16-KB page</td></tr> <tr><td>7</td><td>32K</td><td>32-KB page</td></tr> <tr><td>8</td><td>64K</td><td>64-KB page</td></tr> <tr><td>9</td><td>128K</td><td>128-KB page</td></tr> <tr><td>10</td><td>256K</td><td>256-KB page</td></tr> <tr><td>11</td><td>512K</td><td>512-KB page</td></tr> <tr><td>12</td><td>1M</td><td>1-MB page</td></tr> </tbody> </table>	Value	Name	Description	0	256	256-byte page	1	512	512-byte page	2	1K	1-KB page	3	2K	2-KB page	4	4K	4-KB page	5	8K	8-KB page	6	16K	16-KB page	7	32K	32-KB page	8	64K	64-KB page	9	128K	128-KB page	10	256K	256-KB page	11	512K	512-KB page	12	1M	1-MB page					
Value	Name	Description																																															
0	256	256-byte page																																															
1	512	512-byte page																																															
2	1K	1-KB page																																															
3	2K	2-KB page																																															
4	4K	4-KB page																																															
5	8K	8-KB page																																															
6	16K	16-KB page																																															
7	32K	32-KB page																																															
8	64K	64-KB page																																															
9	128K	128-KB page																																															
10	256K	256-KB page																																															
11	512K	512-KB page																																															
12	1M	1-MB page																																															
15–14	RESERVED		RO	0	–	–	X	X																																									
13	MEM_INIT	Initializes memories with a hardware state machine. No context accesses should be done while this bit is set. Bit clears when initialization is done. Current usage is to initialize usage count memory to zeroes. No other memories are initialized.	RW	0x1	–	–	X	X																																									
12–8	FLUSH_AHEAD	These bits control how many entries are flushed out of the cache ahead of time to allow reads to immediately be issued. Valid settings are 0–31.	RW	0x10	–	–	X	X																																									
7–4	RESERVED		RO	0	–	–	X	X																																									
3	DISABLE_COMBINE_READ	When set, no block reads will be combined and all the DMAE read request will be 64B.			–	–	X	X																																									
2	DISABLE_PLRU	When set, this bit forces all of the recently accessed bits to zero such that the PLRU algorithm is effectively disabled.	RW	0	–	–	X	X																																									
1	DISABLE_USAGE_CNT	When set, this bit disables the checking of the usage count such that any block can be replaced in the cache.	RW	0	–	–	X	X																																									

Table 169: CTX Command Register (ctx_command, Offset 0x1000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	ENABLED	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the block is enabled (from the central Enable control registers). Writing this bit as a 0 has no effect. This bit is controlled by the “ MISC Enable Set Register (misc_enable_set_bits, Offset 0x810) ” on page 232. When this bit is cleared, the Context block should remove the grant from the current user (if there is one) as if a higher priority requester is active. No new requestors should be serviced.	RO	0	X	X	X	X

CTX STATUS REGISTER (CTX_STATUS, OFFSET 0x1004)

Table 170: CTX Status Register (ctx_status, Offset 0x1004)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–20	RESERVED		RO	0	X	X	–	–
31–27	RESERVED	–	RO	0	–	–	X	X
26	INVALID_PAGE	This bit is set when an attempt to access data from a page where the valid bit of the page table entry is 0. It is also set when page size is less than 4K and aliasing of the address occurs. In any case, external host access will not be performed. When set, the attn signal to HC is set. This bit can be cleared by writing a '1' to it.	WC	0	–	–	X	X
25	USAGE_CNT_ERR	This bit is set when an error occurs with the usage count mechanism. More information about the error can be found in the rep. status register. When set, the attn signal to HC is set. This bit can be cleared by writing a '1' to it.	WC	0	–	–	X	X
24	DEAD_LOCK	This bit is set when the cache is completely full and locked down and 328 µs have elapsed without being able to do a replacement. When set, the attn signal to HC is set. This bit can be cleared by writing a '1' to it.	WC	0	–	–	X	X
23	HIT_STAT	This bit is the current value of the CTX HIT STAT. This bit will toggle once each time there have been 64 hits on the CAM. This generic stat must be compared to the system wide core clock counting statistic to determine the rate of hits on the CAM.	RO	0	–	–	X	X
22	MISS_STAT	This bit is the current value of the CTX MISS STAT. This bit will toggle once each time there have been 64 misses on the CAM. This generic stat must be compared to the system wide core clock counting statistic to determine the rate of hits on the CAM.	RO	0	–	–	X	X
21	EXT_WRITE_STAT	This bit is the current value of the generic CTX EXT WRITE STAT. This bit will toggle once each time 16 blocks of context are written to external memory. This generic stat must be compared to the system wide Core clock counting statistic to determine the rate of external write operations.	RO	0	–	–	X	X

Table 170: CTX Status Register (ctx_status, Offset 0x1004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
20	EXT_READ_STAT	This bit is the current value of the generic CTX EXT READ STAT. This bit will toggle once each time 16 blocks of context are read from external memory. This generic stat must be compared to the system wide Core clock counting statistic to determine the rate of external read operations.	RO	0	-	-	X	X
19	LOCK_STALL_STAT	This bit is the current value of the generic CTX_LOCK_STALL_CLKS. This bit will toggle once each time there have been 128 total CPU clocks of stall experienced by the user blocks for requesting or freeing of locks. If more than one block is stalled for any one clock, the statistics accounts (i.e., add) for the stall experienced by each block. This generic stat must be compared to the system wide Core clock counting statistic to determine the rate of access stalls on the context.	RO	0	X	X	X	X
18	ACC_STALL_STAT	This bit is the current value of the generic CTX_ACC_STALL_STAT. This bit will toggle once each time there have been 128 total CPU clocks of stall experienced by the user blocks for reads or writes. If more than one block is stalled for any one clock, the statistics accounts (i.e., add) the stall experienced by each block. Each user block stalls one CPU clock, regardless, and the READ_STAT and WRITE_STAT values must be taken into account to analyze this statistic. This generic stat must be compared to the system wide Core clock counting statistic to determine the rate of access stalls on the context.	RO	0	X	X	X	X
17	WRITE_STAT	This bit is the current value of the generic CTX_WRITE_STAT. This bit will toggle once each time there have been 64 32-bit word write to the context. This generic stat must be compared to the system wide Core clock counting statistic to determine the rate of write operations on the context.	RO	0	X	X	X	X

Table 170: CTX Status Register (ctx_status, Offset 0x1004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
16	READ_STAT	This bit is the current value of the generic CTX_READ_STAT. This bit will toggle once each time there have been 64 32-bit word reads from the context. This generic stat must be compared to the system wide Core clock counting statistic to determine the rate of read operations on the context.	RO	0	X	X	X	X
15–1	RESERVED		RO	0	X	X	X	X
0	LOCK_WAIT	This bit indicates that the context lock mechanism is currently blocking one or more users.	RO	0	X	X	X	X

CTX VIRTUAL ADDRESS REGISTER (CTX_VIRT_ADDR, OFFSET 0x1008)*Table 171: CTX Virtual Address Register (ctx_virt_addr, Offset 0x1008)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–21	RESERVED		RO	0	X	X	X	X
20–6	VIRT_ADDR	This value is used as the context virtual address for lookup page programming and reading. This value must be set before the PAGE_TBL field of the “CTX Page Table Register (ctx_page_tbl, Offset 0x100c) on page 297 is read or written.	RW	0	X	X	–	–
20–6	RESERVED		RO	0	–	–	X	X
5–0	RESERVED		RO	0	X	X	X	X

CTX PAGE TABLE REGISTER (CTX_PAGE_TBL, OFFSET 0x100c)*Table 172: CTX Page Table Register (ctx_page_tbl, Offset 0x100c)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–20	RESERVED		RO	0	X	X	X	X
19–6	PAGE_TBL	When this value is written, the data will write to the page table at the address specified in VIRT_ADDR field of the “CTX Virtual Address Register (ctx_virt_addr, Offset 0x1008)” on page 297. This will cause accesses to the specified virtual CID address to map to the physical context page defined in the page table. Reads from this value will return the current page table entry for a virtual page.	RW	0	X	X	–	–
19–6	RESERVED		RO	0	–	–	X	X
5–0	RESERVED		RO	0	X	X	X	X

CTX DATA ADDRESS REGISTER (CTX_DATA_ADR, OFFSET 0x1010)*Table 173: CTX Data Address Register (ctx_data_adr, Offset 0x1010)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–21	RESERVED		RO	0	X	X	X	X
20–2	DATA_ADR	This value is used during accesses to the “CTX Data Register (ctx_data, Offset 0x1014)” on page 298. This value will be interpreted as a virtual address to the context and will be dereferenced to the context through the page table as any normal context access would be. This value must be set before the ctx_data register is accessed.	RW	0	X	X	X	X



Table 173: CTX Data Address Register (ctx_data_adr, Offset 0x1010) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1–0	RESERVED		RO	0	X	X	X	X

CTX DATA REGISTER (CTX_DATA, OFFSET 0x1014)

Description	Mode	Reset	06	08	09	16
When this register is written, the value is written to the context at the virtual context address specified in the "CTX Data Address Register (ctx_data_adr, Offset 0x1010)" on page 297. This write will be completed as a full 32-bit write. No smaller writes are supported through this interface. This write will be completed as a low-priority access. Reads from this register will return the data from the context.	RW	0	X	X	X	X

CTX LOCK REGISTER (CTX_LOCK, OFFSET 0x1018)

This register is used to allow locking of context by a single entity over the GRC bus. This may be needed for initialization of the chip as well as for work-arounds.

Table 174: CTX Lock Register (ctx_lock, Offset 0x1018)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	REQ	When this bit is written as a 1, the lock context interface will request the selected lock activity (as indicated in the LOCK_MODE field) for the CID value selected. The bit will read as 1 until the lock is granted. At that point, the LOCK_REQ bit will read as 0 and the LOCK_STATUS bit will be valid. Writing this bit as a 0 has no effect.	SC	0	X	X	X	X
30	STATUS	This bit indicates the status of the last request operation. This bit is only valid when the LOCK_REQ bit is low. If this bit reads as 1, then the last request was successful. If this bit reads as 0, then the last request failed.	RO	0	X	X	X	X

Table 174: CTX Lock Register (ctx_lock, Offset 0x1018) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16											
29–27	MODE	This field determines the action to be taken when the LOCK_REQ bit is set. This field is ignored if the LOCK_REQ bit is not being written as a 1.	RW	0	X	X	X	X											
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>UNLOCK</td> <td>This mode will free all locks on the specified CID value on this lock channel.</td> </tr> <tr> <td>0x1</td> <td>IMMEDIATE</td> <td>This mode will request a lock, but not block until the lock is granted. The LOCK_STATUS bit must be checked after the LOCK_REQ bit goes back to 0 to determine if the lock was attained.</td> </tr> <tr> <td>0x2</td> <td>SURE</td> <td>This mode will request a lock and wait until the lock is attained. When this mode is used, the lock will be granted when the LOCK_REQ bit returns to 0.</td> </tr> </tbody> </table>	Value	Name	Description	0x0	UNLOCK	This mode will free all locks on the specified CID value on this lock channel.	0x1	IMMEDIATE	This mode will request a lock, but not block until the lock is granted. The LOCK_STATUS bit must be checked after the LOCK_REQ bit goes back to 0 to determine if the lock was attained.	0x2	SURE	This mode will request a lock and wait until the lock is attained. When this mode is used, the lock will be granted when the LOCK_REQ bit returns to 0.					
Value	Name	Description																	
0x0	UNLOCK	This mode will free all locks on the specified CID value on this lock channel.																	
0x1	IMMEDIATE	This mode will request a lock, but not block until the lock is granted. The LOCK_STATUS bit must be checked after the LOCK_REQ bit goes back to 0 to determine if the lock was attained.																	
0x2	SURE	This mode will request a lock and wait until the lock is attained. When this mode is used, the lock will be granted when the LOCK_REQ bit returns to 0.																	
26	GRANTED	This bit indicates if any locks are allocated on this context lock interface and is intended for debug use. If any locks have been attained, this bit will be set. When a successful unlock has been executed, this bit will be cleared. This differs from LOCK_STATUS in that it indicates when locks are in use rather than when lock requests (both free and lock) are complete.	RO	0	X	X	X	X											
25–21	RESERVED		RO	0	X	X	X	X											
20–7	CID_VALUE	This value selects the context that the lock will operate on.	RW	0	X	X	X	X											
6–3	RESERVED		RO	0	X	X	X	X											

Table 174: CTX Lock Register (ctx_lock, Offset 0x1018) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2–0	TYPE	These bits specify the lock types being requested when the LOCK_REQ bit is written as a 1 and the LOCK_MODE is non-0. On all other writes, this field is ignored. Each bit of this field indicates desire to lock a particular resource.	RW	0	X	X	X	X
Value Name Description								
	0x0 VOID	No bits are set in this value, so this is an invalid value for a lock request type.						
	0x1 PROTOCOL	Bit 0 indicates lock request for the TX protocol_flags variable. This lock may coexist with a TX lock or Timer clock of the same CID.						
	0x2 TX	Bit one indicates lock request for the TX Pipeline variables. This lock may coexist with a Timer lock or protocol clock for the same CID.						
	0x4 TIMER	Bit 0 indicates lock request for the Timer variables. This lock may coexist with a TX lock or the protocol lock of the same CID.						
	0x7 COMPLETE	Bit 0 indicates lock for the entire context value indicated. This lock has priority over all other locks and cannot coexist with locks of any other type.						

CTX CONTROL REGISTER (CTX_CTRL, OFFSET 0x101C)**Table 175: CTX Control Register (ctx_ctrl, offset 0x101c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	READ REQ	This bit is set to initiate a read from context with the parameters specified with the other bits in this register. The bit stays high until the access is complete, including any pre-fetch. A subsequent context access should only be attempted once this bit is clear.	SC	0	—	—	X	X

Table 175: CTX Control Register (ctx_ctrl, offset 0x101c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
30	WRITE REQ	This bit is set to initiate a write to context with the parameters specified with the other bits in this register. The bit stays high until the access is complete, including any pre-fetch. A subsequent context access should only be attempted once this bit is clear.	SC	0	-	-	X	X
29–27	RESERVED		RO	0	-	-	X	X
26	ATTR	Ignored on when READ REQ is set. When this bit is set when WRITE REQ is set, no external memory access is done before writing the data to the cache. When ATTR is set on multiple successive write accesses to the same block, care must be taken to ensure that previous writes are not corrupted. This can be done by only setting ATTR on the first access to the block or by locking the block in the cache before doing any writes and then unlocking it when done.	RW	0	-	-	X	X
25–24	PREFETCH SIZE	The number of blocks to pre-fetch when READ REQ or WRITE REQ is set.	RW	0	-	-	X	X
23	NO RAM ACC	Set to indicate not to access the cache memory when READ REQ or WRITE REQ is set. Should be set when modifying the usage count or doing a pre-fetch only.	RW	0	-	-	X	X
22–21	MOD USAGE CNT	The action to take with the block's usage count when READ REQ or WRITE REQ is set.	RW	0	-	-	X	X
20–2	CTX ADDR	The context address to access when READ REQ or WRITE REQ is set. This value is the 32-bit address of the word to access in context.	RW	0	-	-	X	X
1–0	RESERVED		RO	0	-	-	X	X

CTX CTX DATA REGISTER (CTX_CTX_DATA, OFFSET 0x1020)

Description	Mode	Reset	06	08	09	16
This register is used in conjunction with the "CTX Control Register (ctx_ctrl, offset 0x101c)" on page 300 to provide read and write access to context memory. Only 32 bit accesses are possible through the GRC interface.	RW	0	-	-	X	X

CTX ACCESS STATUS REGISTER (CTX_ACCESS_STATUS, OFFSET 0x1040)

Table 176: CTX Access Status Register (ctx_access_status, Offset 0x1040)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	-	-

Table 176: CTX Access Status Register (ctx_access_status, Offset 0x1040) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–10	REQUEST	This field indicates the requests from each master.	RO	0	—	—	X	X
27–17	QUALIFIED_REQUEST	Qualified Request Each bit indicates a qualified request from each of the masters	RO	0	X	X	—	—
16	RESERVED		RO	0	X	X	—	—
15–14	ACCESSMEMORYINITSM	These bits indicate the Access Memory Initialization State Machine.	RO	0	X	X	—	—
13–12	PAGETABLEINITSM	These bits indicate the Page Table Initialization State Machine.	RO	0	X	X	—	—
11–10	ACCESSMEMORYSM	These bits indicate the Access Memory State machine.	RO	0	X	X	—	—
9–4	RESERVED		RO	0	X	X	—	—
9–5	CACHEMASTERENCODED	This field indicates the current master of the cache memory in encoded form.	RO	0x1f	—	—	X	X
4–0	CAMMATERENCODED	This field indicates the current master of the CAM in encoded form.	RO	0x1f	—	—	X	X
3–0	MASTERENCODED	These bits indicate the current master of the context in encoded form.	RO	0	X	X	—	—

CTX DEBUG LOCK STATUS REGISTER (CTX_DBG_LOCK_STATUS, OFFSET 0x1044)**Table 177: CTX Debug Lock Status Register (ctx_dbg_lock_status, Offset 0x1044)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	MATCH	These bits indicate if there is a match in the current lock requests.	RO	0	X	X	X	X
21–10	RESERVED		RO	0	X	X	X	X
9–0	SM	These bits indicate the Lock interface state machine for each master.	RO	0	X	X	X	X

CTX CACHE CONTROL STATUS REGISTER (CTX_CACHE_CTRL_STATUS, OFFSET 0x1048)**Table 178: CTX Cache Control Status Register (ctx_cache_ctrl_status, 0x1048)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31–30	RESERVED		RO	0	—	—	X	X
29	RD_CHAN10_ACTIVE	This indicates read completion for DMA read channel 10 is expected.	RO	0	—	—	X	X
28	RD_CHAN9_ACTIVE	This indicates read completion for DMA read channel 9 is expected.	RO	0	—	—	X	X
27	RD_CHAN8_ACTIVE	This indicates read completion for DMA read channel 8 is expected.	RO	0	—	—	X	X
26	RD_CHAN7_ACTIVE	This indicates read completion for DMA read channel 7 is expected.	RO	0	—	—	X	X

Table 178: CTX Cache Control Status Register (ctx_cache_ctrl_status, 0x1048) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25	RD_CHAN6_ACTIVE	This indicates read completion for DMA read channel 6 is expected.	RO	0	-	-	X	X
24	RD_CHAN5_ACTIVE	This indicates read completion for DMA read channel 5 is expected.	RO	0	-	-	X	X
23	RD_CHAN4_ACTIVE	This indicates read completion for DMA read channel 4 is expected.	RO	0	-	-	X	X
22	RD_CHAN3_ACTIVE	This indicates read completion for DMA read channel 3 is expected.	RO	0	-	-	X	X
21	RD_CHAN2_ACTIVE	This indicates read completion for DMA read channel 2 is expected.	RO	0	-	-	X	X
20	RD_CHAN1_ACTIVE	This indicates read completion for DMA read channel 1 is expected.	RO	0	-	-	X	X
19	RD_CHAN0_ACTIVE	This indicates read completion for DMA read channel 0 is expected.	RO	0	-	-	X	X
18–13	CACHE_ENTRY_NEEDED	This indicates the total number of cache entries needed to fulfill all of the outstanding DMA read requests.	RO	0	-	-	X	X
12–7	FREE_ENTRY_CNT	This indicates how many entries are in the free list. It is not valid until FLUSH START is set.	RO	0	-	-	X	X
6	FLUSH_START	This indicates flushing has started.	RO	0	-	-	X	X
5–2	RESERVED		RO	0	-	-	X	X
1	INVALID_READ_COMP	This bit is set when a read completion happens without a request. This suggests a bug in the CTX block. This bit can only be cleared by reset.	RO	0	-	-	X	X
0	RFIFO_OVERFLOW	This bit is set when the read FIFO for DMAE read request overflows. This suggests a bug in the CTX block. This bit can only be cleared by reset.	RO	0	-	-	X	X

CTX CACHE CONTROL STATE MACHINE STATUS REGISTER (CTX_CACHE_CTRL_SM_STATUS, OFFSET 0x104C)

Table 179: CTX Cache Control State Machine Status Register (ctx_cache_ctrl_sm_status, Offset 0x104c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	-	-
31	RESERVED		RO	0	-	-	X	X
30–16	INVALID_BLK_ADDR	This block address is associated with the last invalid page table entry. This field is only valid when the INVALID_PAGE bit "CTX Status Register (ctx_status, Offset 0x1004)" on page 294 is set. It can only be cleared by reset.	RO	0	-	-	X	X
15–12	RESERVED		RO	0	-	-	X	X

Table 179: CTX Cache Control State Machine Status Register (ctx_cache_ctrl_sm_status, Offset 0x104c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11–9	CS_RFIFOC	This indicates the current state of the read FIFO controller.	RO	0	—	—	X	X
8–6	CS_RTAGC	This indicates the current state of the read completion controller.	RO	0	—	—	X	X
5–3	CS_WFIFOC	This indicates the current state of the write FIFO controller.	RO	0	—	—	X	X
2–0	CS_DWC	This indicates the current state of the DMAE write controller.	RO	0	—	—	X	X

CTX CACHE STATUS REGISTER (CTX_CACHE_STATUS, OFFSET 0x1050)**Table 180: CTX Cache Status Register (ctx_cache_status, Offset 0x1050)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–0	RESERVED		RO	0	X	X	—	—
25–16	MAX_HELD_ENTRIES	The maximum number of non-zero usage count entries that have ever been in the cache at one time (since reset). This value is cleared by writing any value to this register.	WC	0	—	—	X	X
15–10	RESERVED		RO	0	—	—	X	X
9–0	HELD_ENTRIES	The number of non-zero usage count entries that are currently in the cache.	RO	0	—	—	X	X

CTX DMA STATUS REGISTER (CTX_DMA_STATUS, OFFSET 0x1054)**Table 181: CTX DMA Status Register (ctx_dma_status, Offset 0x1054)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	RESERVED		RO	0	X	X	X	X
21–0	RESERVED		RO	0	X	X	—	—
21–20	RD_CHAN10_STATUS	A non-zero value indicates PCIE error on DMA read channel 10. Once the first error is captured, the field can only be cleared by reset.	RO	0	—	—	X	X
19–18	RD_CHAN9_STATUS	A non-zero value indicates PCIE error on DMA read channel 9. Once the first error is captured, the field can only be cleared by reset.	RO	0	—	—	X	X
17–16	RD_CHAN8_STATUS	A non-zero value indicates PCIE error on DMA read channel 8. Once the first error is captured, the field can only be cleared by reset.	RO	0	—	—	X	X

Table 181: CTX DMA Status Register (ctx_dma_status, Offset 0x1054) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–14	RD_CHAN7_STATUS	A non-zero value indicates PCIE error on DMA read channel 7. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
13–12	RD_CHAN6_STATUS	A non-zero value indicates PCIE error on DMA read channel 6. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
11–10	RD_CHAN5_STATUS	A non-zero value indicates PCIE error on DMA read channel 5. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
9–8	RD_CHAN4_STATUS	A non-zero value indicates PCIE error on DMA read channel 4. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
7–6	RD_CHAN3_STATUS	A non-zero value indicates PCIE error on DMA read channel 3. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
5–4	RD_CHAN2_STATUS	A non-zero value indicates PCIE error on DMA read channel 2. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
3–2	RD_CHAN1_STATUS	A non-zero value indicates PCIE error on DMA read channel 1. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X
1–0	RD_CHAN0_STATUS	A non-zero value indicates PCIE error on DMA read channel 0. Once the first error is captured, the field can only be cleared by reset.	RO	0	–	–	X	X

CTX REPORT STATUS REGISTER (CTX_REPORT_STATUS, OFFSET 0x1058)*Table 182: CTX Report Status Register (ctx_report_status, Offset 0x1058)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–19	RESERVED		RO	0	X	X	X	X
18–0	RESERVED		RO	0	X	X	–	–
18	USAGE_CNT_MISS_ERR	An attempt was made to increment a usage count that was already at its maximum value. The client id and entry of the 64B block that had this error is captured. This bit is cleared by writing a'1' to it.	WC	0	–	–	X	X
17	USAGE_CNT_MIN_ERR	An attempt was made to decrement a usage count that was already at a value of 0. The client id and entry of the 64B block that had this error is captured. This bit is cleared by writing a'1' to it.	WC	0	–	–	X	X
16	USAGE_CNT_MAX_ERR	An attempt was made to decrement the usage count for a block that was not in the cache. The client id and entry of the 64B block that had this error is captured. This bit is cleared by writing a'1' to it.	RO	0	–	–	X	X
15	RESERVED		RO	0	–	–	X	X
14–10	ERROR_CLIENT_ID	The client that is associated with the first usage count error. This field is only valid when one of the error bits in this register is set. It can only be cleared by reset.	RO	0	–	–	X	X
Value	Name	Description						
0	MQ	Mailbox Queue						
1	COM0	Completion Processor port 0						
2	COM1	Completion Processor port 1						
3	TXP0	TX Processor port 0						
4	TXP1	TX Processor port 1						
5	RXP0	RX Processor port 0						
6	RXP1	RX Processor port 1						
7	TDMA	TX DMA Engine						

Table 182: CTX Report Status Register (ctx_rep_status, Offset 0x1058) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
14–10 (cont.)	ERROR_CLIENT_ID	8	CP0	Command Processor port 0	RO	0	–	–	X	X
		9	CP1	Command Processor port 1						
		10	TPAT	TX Patch-Up Processor						
		11	RDMA	RX DMA Engine						
		12	RV2P	RX V2P Processor						
		13	TSCH0	TX Scheduler port 0						
		14	TBDR	TX BD Read						
		15	RXPQ	RX Processor Main FTQ						
		16	CPQ	Command Processor Main FTQ						
		17	COMTQ	Completion Processor Timer FTQ						
		18	RV2PMQ	RX V2P Mailbox FTQ						
		19	TSCH1	TX Scheduler port 1						
		20	GRC	General Register Control						
		21	TIMER	Timer						
9–0	ERROR_ENTRY	The cache entry that is associated with the first usage count error. This field is only valid when one of the error bits in this register is set. It can only be cleared by reset.			RO	0	–	–	X	X

CTX CHECKSUM ERROR STATUS REGISTER (CTX_CKSUM_ERROR_STATUS, OFFSET 0x105c)*Table 183: CTX Checksum Error Status Register (ctx_cksum_error_status, Offset 0x105c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–16	EXPECTED	This is the expected checksum for the PCIE data. This is the checksum that is sent by the block that provides the data.	RO	0	–	–	X	X
15–0	CALCULATED	This is the calculated checksum for the PCIE data. This is the checksum that is calculated locally over the data.	RO	0	–	–	X	X

CTX CHANNEL LOCK STATUS 0 REGISTER (CTX_CHNL_LOCK_STATUS_0, OFFSET 0x1080)

This array of registers has the status for lock channel 0. The client is Completion Processor port 0.

Table 184: CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–18	RESERVED		RO	0	X	X	X	X
17	RESERVED		RO	0	X	X	–	–
17–15	TYPE	Current Lock Type.	RO	0	–	–	X	X
16	MODE	When 1, corresponding channel has a lock. When 0, corresponding channel does not have a lock	RO	0	X	X	–	–
15–14	TYPE	Current Lock Type.	RO	0	X	X	–	–
14	MODE	When 1, corresponding channel has a lock. When 0, corresponding channel does not have a lock	RO	0	–	–	X	X
13–0	CID	Current CID if the channel is locked This array of registers has one word for each lock channel and reports the status of that lock channel.	RO	0	X	X	X	X

CTX CHANNEL LOCK STATUS 1 REGISTER (CTX_CHNL_LOCK_STATUS_1, OFFSET 0x1084)

Description	Mode	Reset	06	08	09	16
See the definition for “ CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080) ” on page 308. The client is RX Processor port 0.	RO	0	X	X	X	X

CTX CHANNEL LOCK STATUS 2 REGISTER (CTX_CHNL_LOCK_STATUS_2, OFFSET 0x1088)

Description	Mode	Reset	06	08	09	16
See the definition for “ CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080) ” on page 308. The client is TX Processor port 0.	RO	0	X	X	X	X



CTX CHANNEL LOCK STATUS 3 REGISTER (CTX_CHNL_LOCK_STATUS_3, OFFSET 0x108C)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
See the definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308.	RO	0	X	X	-	-

CTX CHANNEL LOCK STATUS 4 REGISTER (CTX_CHNL_LOCK_STATUS_4, OFFSET 0x1090)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
See the definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308. The client is TX DMA Engine.	RO	0	X	X	X	X

CTX CHANNEL LOCK STATUS 5 REGISTER (CTX_CHNL_LOCK_STATUS_5, OFFSET 0x1094)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
See the definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308. The client is Command Processor port 0.	RO	0	X	X	X	X

CTX CHANNEL LOCK STATUS 6 REGISTER (CTX_CHNL_LOCK_STATUS_6, OFFSET 0x1098)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
See definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308. The client is TX Patchup Processor.	RO	0	X	X	X	X

CTX CHANNEL LOCK STATUS 7 REGISTER (CTX_CHNL_LOCK_STATUS_7, OFFSET 0x109C)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
See definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308. The client is TX Scheduler port 0.	RO	0	X	X	X	X

CTX CHANNEL LOCK STATUS 8 REGISTER (CTX_CHNL_LOCK_STATUS_8, OFFSET 0x10A0)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
See definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308. The Client is Mailbox Queue.	RO	0	X	X	X	X

CTX Channel Lock Status 9 Register (ctx_chnl_lock_status_9, Offset 0x10a4)

Description	Mode	Reset	06	08	09	16
See definition for "CTX Channel Lock Status 0 Register (ctx_chnl_lock_status_0, Offset 0x1080)" on page 308. The client is GRC.	RO	0	-	-	X	X

CTX DEBUG STATE MACHINE REGISTER (CTX_DEBUG_SM, OFFSET 0x10A8)

Description	Mode	Reset	06	08	09	16
This register shows all of the FSM in the context block.	RO	0	-	-	X	X

CTX CACHE CONTROL REGISTER (CTX_CACHE_CTRL, OFFSET 0x10c0)*Table 185: CTX Cache Control Register (ctx_cache_ctrl, offset 0x10c0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	RESERVED		RO	0	X	X	-	-
31	READ_REQ	This bit is set to initiate a read directly from the cache at CACHE_ADDR. The bit stays high until the access is complete. A subsequent cache access should only be attempted once this bit is clear.	SC	0	-	-	X	X
30	WRITE_REQ	This bit is set to initiate a write directly to the cache at CACHE_ADDR. The bit stays high until the access is complete. A subsequent cache access should only be attempted once this bit is clear.	SC	0	-	-	X	X
29-16	RESERVED		RO	0	-	-	X	X
15-2	CACHE_ADDR	The address of the 32-bit word to access in the cache memory.	RW	0	-	-	X	X
1-0	RESERVED		RO	0	-	-	X	X

CTX CACHE DATA REGISTER (CTX_CACHE_DATA, OFFSET 0x10c4)

Description	Mode	Reset	06	08	09	16
This register returns the data to write to the cache or the data read from the cache when an access is made through the "CTX Cache Control Register (ctx_cache_ctrl, offset 0x10c0)" on page 310. The full 32 bits of data are transferred.	RW	0	X	X	X	X

CTX HOST PAGE TABLE CONTROL REGISTER (CTX_HOST_PAGE_TBL_CTRL, OFFSET 0x10c8)*Table 186: CTX Host Page Table Control Register (ctx_host_page_tbl_ctrl, offset 0x10c8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	RESERVED		RO	0	X	X	-	-

Table 186: CTX Host Page Table Control Register (ctx_host_page_tbl_ctrl, offset 0x10c8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	READ_REQ	This bit is set to initiate a read from the page table memory. The bit stays high until the access is complete. A subsequent access should only be attempted once this bit is clear.	SC	0	-	-	X	X
30	WRITE_REQ	This bit is set to initiate a write to the host page table memory. The bit stays high until the access is complete. A subsequent access should only be attempted once this bit is clear.	SC	0	-	-	X	X
29–9	RESERVED		RO	0	-	-	X	X
8–0	PAGE_TBL_ADDR	The address to access in the page table memory.	RW	0	-	-	X	X

CTX HOST PAGE TABLE DATA 0 REGISTER (CTX_HOST_PAGE_TBL_DATA0, OFFSET 0x10cc)**Table 187: CTX Host Page Table Data 0 Register (ctx_host_page_tbl_data0, offset 0x10cc)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	-	-
31–8	VALUE	Lower bits [23:0] of the data to be written to/read from the page table when an access is done through the “CTX Host Page Table Control Register (ctx_host_page_tbl_ctrl, offset 0x10c8)” on page 310 . These represent bits [31:8] of the host memory address.	RW	0	-	-	X	X
7–1	RESERVED		RO	0	-	-	X	X
0	VALID	This bit is used to indicate the page table entry is valid.	RW	0	-	-	X	X

CTX HOST PAGE TABLE DATA 1 REGISTER (CTX_HOST_PAGE_TBL_DATA1, OFFSET 0x10d0)

Description	Mode	Reset	06	08	09	16
The upper bits [55:24] of the data to be written to/read from the page table when an access is done through the “CTX Host Page Table Control Register (ctx_host_page_tbl_ctrl, offset 0x10c8)” on page 310 . These represent bits [63:32] of the host memory address.	RW	0	X	X	X	X

ETHERNET MAC (EMAC) REGISTERS

EMAC MODE REGISTER (EMAC_MODE, OFFSET 0x1400)

Table 188: EMAC Mode Register (emac_mode, Offset 0x1400)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	X	X	X	X
20	ACPI_RCVD	This bit is set each time that an ACPI packet is received when the ACPI_ENA field of “ RPM Configuration Register (rpm_config, Offset 0x1808) ” on page 342 is set to 1. This bit is logically ORed with the MPKT_RCVD signal to create the input to the PCI block that generates PME. This bit is cleared when either a 1 is written to it or during Hard Reset. Core Reset has no effect on this bit.	WC	0	X	X	X	X
19	MPKT_RCVD	This bit is set each time that a Magic packet is received when the MPKT is 1. This bit is logically ORed with the ACPI_RCVD signal to create the input to the PCI block that generates PME. This bit is cleared when either a 1 is written to it or during Hard Reset. Core Reset has no effect on this bit.	WC	0	X	X	X	X
18	MPKT	Set this bit to enable Magic packet recognition.	RW	0	X	X	X	X
17–14	RESERVED		RO	0	–	X	–	–
17–12	RESERVED		RO	0	X	–	X	X
13	BOND_OVRD	When this bit is set, the bond ID value that enabled SerDes mode is overridden by the value of the SERDES_MODE bit.	RW	0	–	X	–	–
12	SERDES_MODE	When this bit is set, the onboard 3.125G SerDes block is selected instead on the on-board Copper PHY. This bit has no effect unless the BOND_OVRD bit is also set.	RW	0	–	X	–	–
11	FORCE_LINK	Set this bit to force link on within the MAC. This bit works regardless of any other settings or status of external pins.	RW	0	X	X	X	X
10	EXT_LINK_POL	Set this bit to interpret the external LINK pin as active low rather than active high.	RW	0x1	X	X	X	X
9	MAX_DEFER_DROP_ENA	When this bit is set, any transmit packet deferral that lasts longer than the value defined by JUMBO_ENA bit of the “ EMAC RX MTU Size Register (emac_rx_mtu_size, Offset 0x149c) ” on page 320 is dropped.	RW	0	X	X	X	X
8	TX_BURST	Set this to enable transmit bursting of half-duplex gigabit packets.	RW	0	X	X	X	X
7	TAGGED_MAC_CTL	Set this to enable the acceptance for Tagged MAC control frames. If this bit is not set, untagged MAC control frames (Pause Frames) will be ignored.	RW	0	X	X	X	X
6–5	RESERVED		RO	0	X	–	–	X

Table 188: EMAC Mode Register (emac_mode, Offset 0x1400) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16														
6	RESERVED		RO	0	-	X	X	X														
5	25G_MODE	When this bit is set and SerDes is enabled, the SerDes operates at 2.5G (3.125 GBaud). The PORT field must be set to 0x2 for proper 2.5G operation. When this bit is clear, the SerDes operates in 1G, 100M, or 10M, depending on the value of the PORT field.	RW	0	-	X	X	-														
4	MAC_LOOP	This bit enables MAC loop-back. For this feature to work, the port mode should be set to NONE to select 1/2 the CORE clock speed for both RX and TX pipe clocks. Link should be forced using FORCE_LINK bit above.	RW	0	X	X	X	X														
3-2	PORT	This field controls the MAC interface being used.	RW	0	X	X	X	X														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>NONE</td> <td>This setting is used to ignore all network interface clock signals and use internal clocks instead. This should be done when using MAC loop-back mode or when programming MAC registers while the LINK state is indeterminate. The clock rate used is 50 MHz.</td> </tr> <tr> <td>1</td> <td>MII</td> <td>This setting is used for 10-Mbit and 100-Mbit modes where the PHY is connected to the MAC using an MII interface.</td> </tr> <tr> <td>2</td> <td>GMII</td> <td>This setting is used for 1000-Mbit mode where the PHY (copper or SerDes) is connected to the MAC using a GMII interface.</td> </tr> <tr> <td>3</td> <td>MII_10M</td> <td>This setting is used for 10-Mbit mode where the PHY (Copper or SerDes) is connected at 10-Mbit rate.</td> </tr> </tbody> </table>	Value	Name	Description	0	NONE	This setting is used to ignore all network interface clock signals and use internal clocks instead. This should be done when using MAC loop-back mode or when programming MAC registers while the LINK state is indeterminate. The clock rate used is 50 MHz.	1	MII	This setting is used for 10-Mbit and 100-Mbit modes where the PHY is connected to the MAC using an MII interface.	2	GMII	This setting is used for 1000-Mbit mode where the PHY (copper or SerDes) is connected to the MAC using a GMII interface.	3	MII_10M	This setting is used for 10-Mbit mode where the PHY (Copper or SerDes) is connected at 10-Mbit rate.					
Value	Name	Description																				
0	NONE	This setting is used to ignore all network interface clock signals and use internal clocks instead. This should be done when using MAC loop-back mode or when programming MAC registers while the LINK state is indeterminate. The clock rate used is 50 MHz.																				
1	MII	This setting is used for 10-Mbit and 100-Mbit modes where the PHY is connected to the MAC using an MII interface.																				
2	GMII	This setting is used for 1000-Mbit mode where the PHY (copper or SerDes) is connected to the MAC using a GMII interface.																				
3	MII_10M	This setting is used for 10-Mbit mode where the PHY (Copper or SerDes) is connected at 10-Mbit rate.																				
1	HALF_DUPLEX	This bit should be set to enable collision detection for either the MII or GMII interfaces. When this bit is set, the CSMA/CD state machines will be enabled.	RW	0	X	X	X	X														
0	RESET	This bit is a self clearing reset command for the MAC block.	SC	0	X	X	X	X														

EMAC STATUS REGISTER (EMAC_STATUS, OFFSET 0x1404)**Table 189: EMAC Status Register (emac_status, Offset 0x1404)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	PARITY_ERROR_STATE	This bit indicates that the EMAC block has detected that the parity error condition exists and it is preventing transmit of new packets. If a packet was being transmitted when the condition was entered, it was errored as if the TX FIFO had under-run.	RO	0	X	X	X	X
30–25	RESERVED		RO	0	X	X	X	X
24	AP_ERROR	This bit is set each time the Auto-polling interface needs service. This bit is cleared through the "EMAC MDIO Auto Poll Status Register (emac_mdio_auto_status, Offset 0x14b8)" on page 324.	RO	0	X	X	X	X
23	MI_INT	This bit is set when the MDINT signal from the copper PHY is driven low. This condition must be cleared in the attached PHY that is driving the MDINT pin. This bit is forwarded as an event to the internal processors.	RO	0	X	X	X	X
22	MI_COMPLETE	This bit is set each time the Management Interface transaction has completed. This bit is cleared by writing a 1 to this bit position. This bit is forwarded as an event to the internal processors.	WC	0	X	X	X	X
21–17	RESERVED		RO	0	X	–	–	X
21–19	RESERVED		RO	0	–	X	X	X
18	SERDES_RXCONFIG_IS_0_CHANGE	This bit is set each time the Rx config is 0 output of the SerDes becomes active ('1'). This means that the remote side is trying to start an auto-negotiation. This bit will be set even if auto-negotiation is disabled in the SerDes block. By clearing this bit after turning off auto-negotiation and then mapping to attention, will allow firmware or host to detect when the remote side is trying to execute auto-negotiation. This bit is cleared by writing a'1' to this bit position.	WC	0	–	X	X	–
17	SERDES_RX_CONFIG_IS_0	This bit reflects the state of the Rx config is 0 output of the SerDes. This value is set when a config word of zero is received by the SerDes and is cleared when a config word that is non-zero is received.	RO	0	–	X	X	–
16	SERDES_NXT_PG_CHANGE	This bit is set when a next page change attention occurs (see "EMAC Attention Enable Register (emac_attention_ena, Offset 0x1408)" on page 316).	WC	0	X	X	X	X

Table 189: EMAC Status Register (emac_status, Offset 0x1404) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15	RESERVED		RO	0	X	X	X	X
14	SERDES_AUTONEG_CHANGE	This bit is set when an Auto-Negotiation change attention occurs (see “ EMAC Attention Enable Register (emac_attention_ena, Offset 0x1408) ” on page 316).	WC	0	X	X	X	X
13	SERDES_AUTONEG_COMPLETE	This bit reflects the current auto-negotiation state of the SerDes block. When this bit is: <ul style="list-style-type: none">• Set, it indicates that the auto-negotiation has completed• Clear, it indicates that the auto-negotiation process has never started or is in progress.	RO	0	X	X	X	X
12	LINK_CHANGE	This bit is set each time the LINK bit changes. Writing a 1 clears this bit. This bit is forwarded as an event to the internal processors.	WC	0	X	X	X	X
11	LINK	This bit represents the current link state as selected by the: <ul style="list-style-type: none">• PORT field (see “EMAC Mode Register (emac_mode, Offset 0x1400)” on page 312). If this field is set to MII or GMII and the internal PHY has link, then the LINK bit is set.• AUTO_POLL bit (see “EMAC MDIO Mode Register (emac_mdio_mode, Offset 0x14b4)” on page 323. If this bit is set, then it displays the link status from the MDIO state machine.)• FORCE_LINK bit (“EMAC Mode Register (emac_mode, Offset 0x1400)” on page 312). If this bit is set, then the LINK bit is set. The EXT_LINK_POL bit (see “ EMAC Mode Register (emac_mode, Offset 0x1400) ” on page 312) affects this last source of link only and a value of 1 indicates that the physical interface is linked (when all is configured correctly). This bit is forwarded to the Host Coalescing (HC) block for use in the status block. The edge detect used there is independent of the one provided by LINK_CHANGE.	RO	0	X	X	X	
10–0	RESERVED		RO	0	X	X	X	X

EMAC ATTENTION ENABLE REGISTER (EMAC_ATTENTION_ENA, OFFSET 0x1408)

The bits in this register enable the corresponding status bits in the “[EMAC Status Register \(emac_status, Offset 0x1404\)](#)” on page 314 to generate the MAC attention conditions to the internal processors and/or the host status block.

Table 190: EMAC Attention Enable Register (emac_attention_ena, Offset 0x1408)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–25	RESERVED		RO	0	X	X	X	X
24	AP_ERROR	Setting this bit enables the AP_ERROR status above as a MAC attention condition.	RW	0	X	X	X	X
23	MI_INT	Setting this bit enables the MI_INT status above as a MAC attention condition.	RW	0	X	X	X	X
22	MI_COMPLETE	Setting this bit enables the MI_COMPLETE status above as a MAC attention condition.	RW	0	X	X	X	X
21–19	RESERVED		RO	0	X	X	X	X
18	SERDES_RX_CONFIG_IS_0_CHANGE	Setting this bit enables the SERDES_RX CONFIG IS 0 CHANGE status above as a MAC attention condition.	RW	0	–	X	X	X
18	RESERVED		RO	0	X	–	–	–
17	RESERVED		RO	0	X	X	X	X
16	NXT_PG_CHANGE	Setting this bit enables the NXT_PG_CHANGE status in the EMAC Status Register (see “ EMAC Status Register (emac_status, Offset 0x1404) ” on page 314) as a MAC attention condition.	WC	0	X	X	X	X
15	RESERVED		RO	0	X	X	X	X
14	AUTONEG_CHANGE	Setting this bit enables the AUTONEG_CHANGE status in the EMAC Status Register (see “ EMAC Status Register (emac_status, Offset 0x1404) ” on page 314) as a MAC attention condition.	RW	0	X	X	X	X
13–12	RESERVED		RO	0	X	X	X	X
11	LINK	Setting this bit enables the LINK status above as a MAC attention condition.	RW	0	X	X	X	X
10–0	RESERVED		RO	0	X	X	X	X

EMAC LED REGISTER (EMAC_LED, OFFSET 0x140c)*Table 191: EMAC LED Register (emac_led, Offset 0x140c)*

Bit	Name	Description	Mode	Reset	06	08	09	16														
31	BLNK_RATE_ENA	This bit is set to enable the use of the BLNK_RATE field defined as follows. If this bit is cleared, then the blink rate will be about 16 Hz.	RW	0	X	X	X	X														
30–19	BLNK_RATE	Specifies the period of each blink cycle (on + off) for Traffic LED in milliseconds. Must be a non-0 value. This 12-bit field is reset to 0x040, giving a default blink period of approximately 15.9 Hz.	RW	0x40	X	X	X	X														
18–11	RESERVED		RO	0	X	X	–	–														
18–17	ACTIVITY_SEL	Enumeration:	RW	0	–	–	X	X														
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Use RX DV as the source for activity.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Use RX DV as the source for activity.</td> </tr> <tr> <td>2</td> <td>2</td> <td>Use RX traffic p as the source for activity.</td> </tr> <tr> <td>3</td> <td>3</td> <td>Use RX frame accepted as the source for activity.</td> </tr> </tbody> </table>	Value	Name	Description	0	0	Use RX DV as the source for activity.	1	1	Use RX DV as the source for activity.	2	2	Use RX traffic p as the source for activity.	3	3	Use RX frame accepted as the source for activity.					
Value	Name	Description																				
0	0	Use RX DV as the source for activity.																				
1	1	Use RX DV as the source for activity.																				
2	2	Use RX traffic p as the source for activity.																				
3	3	Use RX frame accepted as the source for activity.																				
16–13	RESERVED		RO	0	–	–	X	X														
12	2500MB_OVERRIDE	If set along with the OVERRIDE bit, turns on the 2500-Mbps LED.	RW	0	–	–	X	X														
11	2500MB	This bit indicates the current status of the 2500-MB LED.	RO	0	–	–	X	X														
10	TRAFFIC_STAT	This bit indicates the current status of the traffic LED.	RO	0	X	X	X	X														
9	10MB	This bit indicates the current status of the 10-MB LED.	RO	0	X	X	X	X														
8	100MB	This bit indicates the current status of the 100-MB LED.	RO	0	X	X	X	X														
7	1000MB	This bit indicates the current status of the 1000-MB LED.	RO	0	X	X	X	X														
6	TRAFFIC	If set along with the TRAFFIC_OVERRIDE bit, turns on the Traffic LED. If the BLNK_TRAFFIC bit is also set, the LED will blink with blink rate specified in BLNK_RATE and BLNK_RATE_ENA fields.	RW	0	X	X	X	X														
5	BLNK_TRAFFIC	If set along with the TRAFFIC_OVERRIDE bit and Traffic LED bit, the Traffic LED will blink with the blink rate specified in BLNK_RATE and BLNK_RATE_ENA fields.	RW	0	X	X	X	X														
4	TRAFFIC_OVERRIDE	If set, overrides hardware control of the Traffic LED. The Traffic LED will then be controlled via bit 6 and bit 5.	RW	0	X	X	X	X														
3	10MB_OVERRIDE	If set along with the OVERRIDE bit, turns on the 10-Mbps LED.	RW	0	X	X	X	X														

Table 191: EMAC LED Register (emac_led, Offset 0x140c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2	100MB_OVERRIDE	If set along with the OVERRIDE bit, turns on the 100-Mbps LED.	RW	0	X	X	X	X
1	1000MB_OVERRIDE	If set along with the OVERRIDE bit, turns on the 1000-Mbps LED.	RW	0	X	X	X	X
0	OVERRIDE	If set, overrides hardware control of the 3 link LEDs using the OVERRIDE bit values above. When this bit is cleared, the 10-MB LED will light when LINK is set in the mdio_status register, MII mode is selected mode register, and 10 MB is set in the mdio_status register. When this bit is cleared, the 100-MB LED will light when LINK is set in the mdio_status register, MII mode is selected mode register, and 10 MB is cleared in the mdio_status register. When this bit is cleared, the 1000-MB LED will light when LINK is set in the mdio_status register and GMII or TBI mode are selected mode register.	RW	0	X	X	X	X

EMAC MAC MATCH REGISTERS (EMAC_MAC_MATCH[32], OFFSET 0x1410)

Description	Mode	Reset	06	08	09	16
These registers contain 16 48-bit perfect match filter values. The even locations RW contain the upper 16 bits of the filter value in bit positions 15 down to 0. The odd registers contain the lower 32-bits of each perfect match filter value. MAC address values can be unicast, multicast, or the broadcast address. When the SORT_MODE field (see “ EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8) ” on page 327) is 0, then all values are valid at all times. Unless the PROMISCUOUS bit is set, all unicast packets that do not match one of the values will be dropped and counted in the filter_discards statistic. If the SORT_MODE bit is 1, then each 48-bit value is disabled by default and must be individually enabled for any one or combination of users using the corresponding PM_EN bit in the RX Parser sorting logic user registers. All unicast packets that do not match any filters enabled for one or more users will be dropped and counted in the filter_discards statistic. Each perfect match address may be individually enabled/disabled through the PM_EN field of the “ RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820) ” on page 345.	0	X	X	X	X	



Note: While the NetXtreme II hardware supports 16 perfect match filter, some of these registers are reserved for firmware operation and are not available for host OS driver developers. At the time this document was written, the perfect match filters have been allocated as follows: 0 – Standard L2 MAC address; 1– Available; 2 – Reserved for iSCSI; 3 – Available; 4 through 7 – Reserved for Broadcom teaming software (i.e. BASP); 8 through 14 – Available; 15 – Reserved for Boradcom management firmware (i.e. IPMI, UMP, or NC-SI).

EMAC BACKOFF SEED REGISTER (EMAC_BACKOFF_SEED, OFFSET 0x1498)

Table 192: EMAC Backoff Seed Register (emac_backoff_seed, Offset 0x1498)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–10	RESERVED		RO	0	X	X	X	X
9–0	EMAC_BACKOFF_SEED	This value is the seed used for the random back-off timer use after each collision. Writing this value will enable the LFSR random generator. This should be done with lower 3 bytes of the first MAC address. The read value of this register is indeterminate. By leaving this value as 0, back-off on collision re-transmit will be disabled. This should only be done for testing.	RW	0	X	X	X	X

EMAC RX MTU SIZE REGISTER (EMAC_RX_MTU_SIZE, OFFSET 0x149c)

Table 193: EMAC RX MTU Size Register (emac_rx_mtu_size, Offset 0x149c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	JUMBO_ENA	When this bit is set, the RX MAC will disable the length out-of-range check where the TYPE/LEN field of the Ethernet header is compared to the packet length. In the TX MAC, this bit, along with the value of the PORT field (see “ EMAC Mode Register (emac_mode, Offset 0x1400) ” on page 312), changes the number of clocks used for the max deferral check according to the following table:	RW	0	X	X	X	X
		PORT = 1 (MII) PORT = 2 (GMII)						
		JUMBO_ENA = 0 3044 clocks 19444 clocks						
		JUMBO_ENA = 1 18044 clocks 34444 clocks						
30-16	RESERVED		RO	0	X	X	X	X
15-0	MTU_SIZE	Any packet received that is longer than this value will be marked as oversized to the RX MAC. The RDE will drop those packets if configured to do so. The length check includes the L2 header and the CRC value, but it does not include any VLAN tags, so this setting is the same regardless of any VLAN operations. The maximum allowable value for this field is 0x233a (9018). Setting a value larger than 0x223a may produce unpredictable results.	RW	0x05ee	X	X	X	X

EMAC MDIO COMMAND REGISTER (EMAC_MDIO_COMM, OFFSET 0x14AC)*Table 194: EMAC MDIO Command Register (emac_mdio_comm, Offset 0x14ac)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–30	RESERVED		RO	0	X	X	X	X
29	START_BUSY	When written to a 1, the currently programmed MDIO transaction will activate. When the operation is complete, this bit will clear and the MI_COMPLETE bit will be set in the emac_status register. Writing this bit as a 0 has no effect. This bit must be read as a 0 before setting to prevent unpredictable results.	SC	0	X	X	X	X
28	FAIL	This bit is updated at the end of each MDIO transaction when the START_BUSY bit is set. If an error occurred on the MDIO interface during the operation, this bit will be updated to 1, otherwise, it will be updated to 0. Errors usually happen when the attached PHY fails to drive a response during a read. This bit is only modified when a new MDIO transaction is completed.	RO	0	X	X	X	X
27–26	COMMAND	This field controls the type of MDIO transaction that will be performed when the START_BUSY bit is set.	RW	0	X	X	–	–
Value Name								
0	UNDEFINED_0	This value is undefined.						
1	WRITE	This value causes a write to be performed. The data in the DATA field is written to the MDIO address specified when the START_BUSY bit is set.						
2	READ	This value causes a read to be performed. A read is done from the MDIO address specified and the results are placed in the DATA field when the START_BUSY bit is set.						
3	UNDEFINED_3	This value is undefined.						

Table 194: EMAC MDIO Command Register (emac_mdio_comm, Offset 0x14ac) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
27–26	COMMAND	This field controls the type of MDIO transaction that will be performed when the START_BUSY bit is set.	RW	0	–	–	X	X
		Value Name	Description					
		0 UNDEFINED_0	This value is undefined.					
		0 ADDRESS	This will execute the address phase of the MDIO transaction.					
		1 WRITE_22	This value causes a Clause 22 write to be performed. The data in the DATA field is written to the MDIO address specified when the START_BUSY bit is set.					
		1 WRITE_45	This value causes a Clause 45 write to be performed. The data in the DATA field is written to the MDIO address specified with the START_BUSY bit is set.					
		2 READ_22	This value will cause a Clause 22 read to be performed. A read will be done from the MDIO address specified and the results will be placed in the DATA field when the START_BUSY bit is set.					
		2 READ_INC_45	This value will cause a Clause 45 read to be performed by incrementing the address written during the address phase. This address gets incremented every time a read operation is performed. This command removes the need for an address phase before every data phase.					
		3 UNDEFINED_0	This value is undefined.					
		3 READ_45	This value will cause a Clause 45 read to be performed. A read will be done from the MDIO address specified and the results will be placed in the DATA field when the START_BUSY bit is set.					
25–21	PHY_ADDR	This value is used to define the PHY address portion of the MDIO transaction. For Clause 45 transactions it specifies the port address.	RW	0x1	X	X	X	X
20–16	REG_ADDR	This value is used to define the register address portion of the MDIO transaction. This selects what register within a PHY device is being accessed. For Clause 45 transactions, this field specifies device address.	RW	0	X	X	X	X
15–0	DATA	When this register is read, it returns the results of the last MDIO transaction that was performed. When this register value is written, it updates the value that will be used on the next MDIO write transaction that will be performed. For Clause 45 transactions this field specifies the register address.	RW	0	X	X	X	X

EMAC MDIO STATUS REGISTER (EMAC_MDIO_STATUS, OFFSET 0x14B0)*Table 195: EMAC MDIO Status Register (emac_mdio_status, Offset 0x14b0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–2	RESERVED		RO	0	X	X	X	X
1	10MB	This bit is manually controlled only. It is not effected at all by the MDIO interface. The value of this setting is used by the LED logic.	RW	0	X	X	X	X
0	LINK	This bit is updated by the MDIO interface if auto-polling is enabled. The value of this bit is reflected in the main link status bit if auto-polling of the MDIO is enabled.	RO	0	X	X	X	X

EMAC MDIO MODE REGISTER (EMAC_MDIO_MODE, OFFSET 0x14B4)*Table 196: EMAC MDIO Mode Register (emac_mdio_mode, Offset 0x14b4)*

Bit	Name	Description	Mode	Reset	06	08	09	09
31–21	RESERVED		RO	0	X	X	—	—
31	CLAUSE_45	When set this bit indicates that the current MDIO transaction will be executed as a Clause 45 transaction. When clear the transaction is executed as a Clause 22 transaction. The value of this bit also determines the meaning of bits [27:0] of the MDIO COMMAND register. This bit must be set to proper value before the link auto-polling function is enabled.	RO	0	—	—	X	X
30–22	RESERVED		RO	0	—	—	X	X
21–16	CLOCK_CNT	This field controls the MDIO clock speed. The output MDIO clock runs at a frequency equal to CORE_CLK/2*CLOCK_CNT. A value of 0 is invalid for this register.	RW	0x13	—	—	X	X
20–16	CLOCK_CNT	This field controls the MDIO clock speed. The output MDIO clock runs at a frequency equal to CORE_CLK/2*CLOCK_CNT. A value of 0 is invalid for this register.	RW	0x13	X	X	—	—
15–13	RESERVED		RO	0	X	—	—	—
15–14	RESERVED		RO	0	—	X	X	X
13	EXT_MDINT	This field reflects the current state of the external MDINT input pin. If the interrupt is asserted, this bit is 0, otherwise it is 1.	RO	X	—	X	X	X

Table 196: EMAC MDIO Mode Register (emac_mdio_mode, Offset 0x14b4) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	09
12	MDINT	The value of this bit reflects the current state of the MDINT input pin from the copper PHY. If the interrupt is asserted, this bit reads as 0, otherwise, it reads as 1.	RO	0x1	X	X	X	X
11	MDC	Setting this bit to 1 will cause the MDC pin to high if the BIT_BANG bit is set. Setting this pin low will cause the MDC pin to drive low if the BIT_BANG bit is set.	RW	0	X	X	X	X
10	MDIO_OE	Setting this bit to 1 will cause the MDIO pin to drive the value written to the MDIO bit if the BIT_BANG bit is set. Setting this bit to 0 will make the MDIO pin an input.	RW	0	X	X	X	X
9	MDIO	The write value of this bit controls the drive state of the MDIO pin if the BIT_BANG bit is set. The read value of this bit always reflects the state of the MDIO pin.	RW	0	X	X	X	X
8	BIT_BANG	If this bit is 1, the MDIO interface is controlled by the MDIO, MDIO_OE, and MDC bits in this register. When this bit is 0, the commands in the mdio_cmd register will be executed.	RW	0	X	X	X	X
7–5	RESERVED		RO	0	X	X	X	X
4	AUTO_POLL	This bit enables auto-polling. When auto-polling is on, the START_BUSY bit in the mdio_comm register must not be set. The interface will automatically poll the PHY device and set the LINK bit in the mdio_status register according to bit 2 of the PHY register 1. The PHY address used is that programmed into the PHY_ADDR field of the mdio_comm register.	RW	0	X	X	X	X
3–2	RESERVED		RO	0	X	X	X	X
1	SHORT_PREAMBLE	If this bit is set, the 32-bit preamble will not be generated during auto-polling.	RW	0	X	X	X	X
0	RESERVED		RO	0	X	X	X	X

EMAC MDIO AUTO POLL STATUS REGISTER (EMAC_MDIO_AUTO_STATUS, OFFSET 0x14B8)**Table 197: EMAC MDIO Auto Poll Status Register (emac_mdio_auto_status, Offset 0x14b8)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–1	RESERVED		RO	0	X	X	X	X
0	AUTO_ERR	This bit is set each time an error is detected during a auto poll sequence. The bit is cleared by writing a 1 to this bit position.	WC	0	X	X	X	X

EMAC TX MODE REGISTER (EMAC_TX_MODE, OFFSET 0x14BC)*Table 198: EMAC TX Mode Register (emac_tx_mode, Offset 0x14bc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–8	RESERVED		RO	0	X	X	X	X
7	LINK_AWARE	When this bit is set, packets will only be transmitted when link is up. Normally, packets are transmitted regardless of the state of link.	RW	0	X	X	X	X
6	LONG_PAUSE	When this bit is set, pause packets will be generated with 0xffff as the value. When this bit is cleared, pause packets will be generated with a value of 0x1fff.	RW	0	X	X	X	X
5	BIG_BACKOFF	When this bit is set, the back-off algorithm will pull one more bit from the LFSR value than is defined by IEEE specification. This means that it will fetch 4 bits of the LFSR after the third collision rather than 3 bits. This will increase the chances of the other end winning.	RW	0	X	X	X	X
4	FLOW_EN	This bit enables the transmission of IEEE 802.3x MAC control pause frames when the receive side of the NIC is congested. The back-off value sent in the pause frame is defined by the LONG_PAUSE bit above. If the NIC becomes uncongested, then a new pause frame will be generated with a back-off value of 0. If the NetXtreme II remains congested until 3/4 of the pause time has expired, a new pause frame will be generated. When this bit is cleared, no pause frames are generated.	RW	0	X	X	X	X
3	EXT_PAUSE_EN	This bit enables the external PAUSE input to stall the transmitter when asserted. This input is normally used by an external in-line processor and provides more efficient flow control than the pause packets enabled by FLOW_EN.	RW	0	X	X	X	X
2–1	RESERVED		RO	0	X	X	–	–
2	CS16_TEST	When this bit is set to '1', the TX MAC inverts one bit in the calculated data integrity checksum causing mismatch with the checksum received from the TAS. Used only for test.	RW	0	–	–	X	X
1	RESERVED		RO	0	–	–	X	X
0	RESET	When this bit is set, the Transmit MAC state machine will be reset. The bit will self clear when the reset is complete.	SC	0	X	X	X	X

EMAC TX STATUS REGISTER (EMAC_TX_STATUS, OFFSET 0x14c0)*Table 199: EMAC TX Status Register (emac_tx_status, Offset 0x14c0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–6	RESERVED		RO	0	X	X	X	X
5	RESERVED		RO	0	X	X	—	—
5	CS16_ERROR	When this bit is set to '1' it indicates that the calculated data integrity checksum mismatched with the checksum received from the TAS.	RO	0	—	—	X	X
4	UNDERRUN	When this bit is set to indicate the TX FIFO has been under-run. This is not a normal condition. The error flag can be cleared by writing a 1 to this bit position.	WC	0	X	X	X	X
3	LINK_UP	When this bit indicates the link status as interpreted by the MAC. It will read as a 1 if link is up.	RO	0	X	X	X	X
2	XON_SENT	This bit is set each time a pause frame with a back-off value of 0 is sent. This bit is cleared by writing a 1 to this bit location.	WC	0	X	X	X	X
1	XOFF_SENT	This bit is set each time a pause frame with a back-off value of 0x1fff is sent. This bit is cleared by writing a 1 to this bit location.	WC	0	X	X	X	X
0	XOFFED	This bit is set during the time between when an XOFF pause packet is sent and the XON pause packet is sent. This bit is set during the time the MAC intends to pause the line partner from sending.	RO	0	X	X	X	X

EMAC TX LENGTHS REGISTER (EMAC_TX_LENGTHS, OFFSET 0x14c4)*Table 200: EMAC TX Lengths Register (emac_tx_lengths, Offset 0x14c4)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–14	RESERVED		RO	0	X	X	X	X
13–12	IPG_CRS	This value controls the number of 16-bit word times after the end of a packet that CRS will be ignored.	RW	0x2	X	X	X	X
11–8	IPG	This value controls the number of 16-bit word times that will be inserted between transmit frames by the transmitter.	RW	0x6	X	X	X	X
7–0	SLOT	This value controls the number of 16-bit word times that will constitute one slot time for the calculation of collision backup.	RW	0x20	X	X	X	X

EMAC RX MODE REGISTER (EMAC_RX_MODE, OFFSET 0x14c8)*Table 201: EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–13	RESERVED		RO	0	X	X	X	X
12	SORT_MODE	This bit should be set when using the sorting logic of the RX Parser. It affects the dropping of packets not selected by the RX Parser logic PM_EN bits. If this bit is 0, then packets will be dropped if they do not hit one of the 16 perfect match filters, one of the 256 MC Hash filter buckets, or are broadcast when the FILT_BROADCAST bit is not set. If this bit is 1, then dropping is automatically selected on a per-packet basis when it is not selected by any of the users in the RX Parser sorting logic.	RW	0	X	X	X	X
11	FILT_BROADCAST	When the SORT_MODE bit is 0, this bit controls if packets with the broadcast Ethernet destination address (all 1s) will be filtered. When the SORT_MODE bit is 1, this bit will have no effect.	RW	0	X	X	X	X
10	KEEP_VLAN_TAG	Setting this bit forces the RX MAC to keep the VLAN tag in the data delivered to the RX MBUF area. This bit should only be set for debugging reasons. This bit affects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
9	NO_CRC_CHK	Setting this bit causes the CRC status of frames to be ignored and all frames to be processed as good frames. It does affect the operation of statistics. This bit affects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
8	PROMISCUOUS	Setting this bit causes the MAC address checking and MC hashing checking to be ignored on incoming frames. All frames will be processed. This bit does not affect the operation of statistics. When the user sorting logic of the RX Parser is in use, this bit should be programmed to 0 to allow the sorting logic to operate correctly. If this bit is set, it will accept all packets, regardless of sorting logic programming. When sorting logic is in use, each user may request promiscuous mode using its register in the RX Parser block.	RW	0	X	X	X	X

Table 201: EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7	LLC_CHK	Setting this bit enables dropping of frames that fail the IEEE 802.3 length check done on LLC frames. The LLC check consists of ensuring that the length field matches the length of the packet unless it's value is more than 0x600. If this bit is 0, then the LLC length field will have to effect on MAC operation. This bit does not affect the operation of statistics. This bit affects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
6	ACCEPT_RUNTS	Setting this bit causes frames of less than 64 bytes to be processed instead of being dropped. This bit does not affect the operation of statistics. This bit effects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
5	ACCEPT_OVERSIZE	Setting this bit causes frames of more than the rx_mtu_size register to be processed. This bit does not affect the operation of statistics. Frames longer than 16384 bytes will be truncated regardless of this setting as they may cause lock-up conditions. This bit affects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
4	KEEP_PAUSE	Setting this bit causes pause frames to be passed on for processing. This setting has no affect on the operation of the pause frames. This bit affects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
3	KEEP_MAC_CONTROL	Setting this bit causes MAC control frames (except for pause frames) to be passed on for processing. This setting has no affect on the operation of the pause frames. This bit affects all packets regardless of RX Parser packet sorting logic.	RW	0	X	X	X	X
2	FLOW_EN	Setting this enables the processing of pause frames by the RX MAC. When enabled, the back-off value from each pause frame received will be loaded into the TX back-off timer and will stall that transmitter until the back-off has expired.	RW	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	RESET	When this bit is set, the Receive MAC state machine will be reset. The bit will self clear when the reset is complete.	SC	0	X	X	X	X

EMAC RX STATUS REGISTER (EMAC_RX_STATUS, OFFSET 0x14CC)*Table 202: EMAC RX Status Register (emac_rx_status, Offset 0x14cc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–3	RESERVED		RO	0	X	X	X	X
2	N_RECEIVED	This bit is set each time a pause packet with a back-off value of 0 is received. This bit is cleared by writing a 1 to this bit location.	WC	0	X	X	X	X
1	FF_RECEIVED	This bit is set each time a pause packet with a back-off value of more than 0 is received. This bit is cleared by writing a 1 to this bit location.	WC	0	X	X	X	X
0	FFED	This bit read as 1 if a the transmitter is currently stopped due to the reception of a pause packet from the link partner.	RO	0	X	X	X	X

EMAC MULTICAST HASH REGISTERS (EMAC_MULTICAST_HASH[8], OFFSET 0x14D0)

Description	Mode	Reset	06	08	09	16
The multicast hash registers are used to allow separate filtering of multicast frames from broadcast frames. The CRC value after the first 6 bytes of the frame (the multicast destination address) have been processed is used as the hash value. The bottom 8 bits of this value are used as the index into the 256 bits provided by these 8 registers. The index progresses from the LSB to MSB of each register starting with emac_multicast_hash[0]. If the corresponding bit is set, then the multicast frame will be processed.	RW	0	X	X	X	X

EMAC CHECKSUM ERROR STATUS REGISTER (EMAC_CKSUM_ERROR_STATUS, OFFSET 0x14F0)*Table 203: EMAC Checksum Error Status Register (EMAC_CKSUM_ERROR_STATUS, Offset 0x14f0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31–16	EXPECTED	This is the expected data integrity checksum. This is the checksum that is sent by the block that provides the data.	RO	0	—	—	X	X
15–0	CALCULATED	This is the calculated data integrity checksum. This is the checksum that is calculated locally over the data.	RO	0	—	—	X	X

EMAC IFHCINOCTETS REGISTER (EMAC_RX_STAT_IFHCINOCTETS, OFFSET 0x1500)

Description	Mode	Reset	06	08	09	16
This register returns the number of octets received on the interface, including framing characters for packets that were accepted by the L2 filters. Only the lower 19 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHCINBADOCTETS REGISTER (EMAC_RX_STAT_IFHCINBADOCTETS, OFFSET 0x1504)

Description	Mode	Reset	06	08	09	16
This register returns the number of octets received on the interface, including framing characters for packets that were dropped in the MAC for any reason. Only the lower 19 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSFragments REGISTER (EMAC_RX_STAT_ETHERSTATSFragments, OFFSET 0x1508)

Description	Mode	Reset	06	08	09	16
This register returns the count of frames less than 64 bytes with bad FCS. Only the lower 19 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHCINUNICASTPKTS REGISTER (EMAC_RX_STAT_IFHCINUNICASTPKTS, OFFSET 0x150c)

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received on the wire that were not dropped due to errors and have Unicast Ethernet destination addresses. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHCINMULTICASTPKTS REGISTER (EMAC_RX_STAT_IFHCINMULTICASTPKTS, OFFSET 0x1510)

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received on the wire that were not dropped due to errors and have multicast Ethernet destination addresses. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHCINBROADCASTPKTS REGISTER (EMAC_RX_STAT_IFHCINBROADCASTPKTS, OFFSET 0x1514)

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received on the wire that were not dropped due to errors and have the broadcast Ethernet destination addresses. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC DOT3STATSFCSERRORS REGISTER (EMAC_RX_STAT_DOT3STATSFCSERRORS, OFFSET 0x1518)

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received on the wire that have an even number of nibbles, fail FCS check, and are of legal length. This statistic will not count due to line FCS errors if the NO_CRC_CHK bit is set (see the "EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8)" on page 327). If the PHY indicates any error bytes (coding errors), then the whole frame will be counted as if it had an FCS error. Only the lower 13 bits of this register count are valid.	RO	0	X	X	X	X

EMAC DOT3STATSALIGNMENTERRORS REGISTER (EMAC_RX_STAT_DOT3STATSALIGNMENTERRORS, OFFSET 0x151c)

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received on the wire that have an odd number of nibbles, fail FCS check, and are of legal length. This statistic will not count due to line FCS errors if the NO_CRC_CHK bit is set (see the "EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8)" on page 327). If the PHY indicates any error bytes (coding errors), then the whole frame will be counted as if it had an FCS error. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC DOT3STATSCARRIERSENSEERRORS REGISTER (EMAC_RX_STAT_DOT3STATSCARRIERSENSEERRORS, OFFSET 0x1520)

Description	Mode	Reset	06	08	09	16
This register returns the number of times a false carrier has been detected on the internal PHY device. This is indicated from the PHY by asserting RXER while RXDV is low when the RXD pins are at a state of 0x0e. Only the lower 19 bits of this register are valid.	RO	0	X	X	X	X

EMAC XONPAUSEFRAMESRECEIVED REGISTER (EMAC_RX_STAT_XONPAUSEFRAMESRECEIVED, OFFSET 0x1524)

Description	Mode	Reset	06	08	09	16
This register returns the number of good MAC control frames received of pause type with a back-off value of 0. This register will increment regardless of the flow control state. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC XOFFPAUSEFRAMESRECEIVED REGISTER (EMAC_RX_STAT_XOFFPAUSEFRAMESRECEIVED, OFFSET 0x1528)

Description	Mode	Reset	06	08	09	16
This register returns the number of good MAC control frames received of pause type with a back-off value other than 0. This register will increment regardless of the flow control state. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

**EMAC MACCONTROLFRAMESRECEIVED REGISTER
(EMAC_RX_STAT_MACCONTROLFRAMESRECEIVED, OFFSET 0x152c)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good MAC control frames received that are not of pause type. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC XOFFSTATEENTERED REGISTER (EMAC_RX_STAT_XOFFSTATEENTERED, OFFSET 0x1530)

Description	Mode	Reset	06	08	09	16
This register returns the number of times the flow control state has been entered due to the reception of an XOFF pause frame when not in the paused state. The pause flow control mode must be enabled for this counter to increment. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

**EMAC DOT3STATSFRAMESTOOLONG REGISTER
(EMAC_RX_STAT_DOT3STATSFRAMESTOOLONG, OFFSET 0x1534)**

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received that exceed the programmed MTU size (see " EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8) " on page 327). Only the lower 10 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSJABBERS REGISTER (EMAC_RX_STAT_ETHERSTATSJABBERS, OFFSET 0x1538)

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received that exceed the programmed MTU size and have bad FCS. If the PHY indicates any error bytes (coding errors), then the whole frame will be counted as if it had an FCS error. Only the lower 10 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSUNDERSIZEPKTS REGISTER
(EMAC_RX_STAT_ETHERSTATSUNDERSIZEPKTS, OFFSET 0x153c)**

Description	Mode	Reset	06	08	09	16
This register returns the number of frames received that are less than 64 bytes in length. Only the lower 18 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS64OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS64OCTETS, OFFSET 0x1540)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received that are 64 bytes in size. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS65OCTETSTo127OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS65OCTETSTO127OCTETS, OFFSET 0x1544)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received that are 65 bytes to 127 bytes in size. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS128OCTETSTo255OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS128OCTETSTO255OCTETS, OFFSET 0x1548)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received that are 128 bytes to 255 bytes in size. Only the lower 11 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS256OCTETSTo511OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS256OCTETSTO511OCTETS, OFFSET 0x154C)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received that are 256 bytes to 511 bytes in size. Only the lower 10 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS512OCTETSTo1023OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS512OCTETSTO1023OCTETS, OFFSET 0x1550)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received of 512 bytes to 1023 bytes in size. Only the lower 9 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS1024OCTETSTo1522OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS1024OCTETSTO1522OCTETS, OFFSET 0x1554)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received of 1024 bytes to 1522 bytes in size. Only the lower 9 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS1523OCTETSTo9022OCTETS REGISTER
(EMAC_RX_STAT_ETHERSTATSPKTS1523OCTETSTO9022OCTETS, OFFSET 0x1558)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames received of 1523 bytes to 9022 bytes in size. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X

**EMAC RECEIVE STATISTICS FALSE CARRIER ERRORS REGISTER
(EMAC_RX_STAT_FALSECARRIERERRORS, OFFSET 0x1574)**

Description	Mode	Reset	06	08	09	16
This register represents the number of times a set of false carrier events has been detected on the internal PHY device. This is indicated from the PHY by asserting RXER while RXDV is low when the RXD pins are at a state of 0x0e.	RO	0	-	B	X	X

EMAC RECEIVE STATISTICS AUTO-CLEAR REGISTERS (EMAC_RX_STAT_AC[23], OFFSET 0x1580)

Description	Mode	Reset	06	08	09	16
These are the auto-clear versions of the RX MAC Statistics registers, “ EMAC AC ifHcInOctets Register (emac_rx_stat_ifhcinoctets, Offset 0x1500) ” on page 329 to “ EMAC etherStatsPkts1523OctetsTo9022Octets Register (emac_rx_stat_etherstatspkts1523octetsto9022octets, Offset 0x1558) ” on page 334. These locations are used by the Host Coalescing block to aggregate statistics into its local memory.	0		X	X	X	X

EMAC RECEIVE STATISTICS REGISTER (EMAC_RX_STAT_AC_28, OFFSET 0x15F4)

Description	Mode	Reset	06	08	09	16
This is the auto-clear version of the FALSECARRIERERRORS statistic. This value is AC not read by the Host Coalescing block.	0	-	B	X	X	

EMAC IFHCOUTOCTETS REGISTER (EMAC_TX_STAT_IFHCOUTOCTETS, OFFSET 0x1600)

Description	Mode	Reset	06	08	09	16
This register returns the number of octets that have been transmitted on the interface. Only the lower 19 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHCOUTBADOCTETS REGISTER (EMAC_TX_STAT_IFHCOUTBADOCTETS, OFFSET



0x1604)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This register returns the number of octets that have been dropped by the transmit mac RO for reasons such as excessive collisions, late collisions, etc. Only the lower 19 bits of this register are valid.	0		X	X	X	X

EMAC ETHERSTATSCOLLISIONS REGISTER (EMAC_TX_STAT_ETHERSTATSCOLLISIONS, OFFSET 0x1608)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This register returns the number of collisions that have been detected on the interface. RO Only the lower 18 bits of this register are valid.	0		X	X	X	X

EMAC OUTXONSENT REGISTER (EMAC_TX_STAT_OUTXONSENT, OFFSET 0x160C)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This register returns the number of MAC Control pause packets with a value 0xffff that RO have be transmitted. Only the lower 13 bits of this register are valid.	0		X	X	X	X

EMAC OUTXOFFSENT REGISTER (EMAC_TX_STAT_OUTXOFFSENT, OFFSET 0x1610)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This register returns the number of MAC Control pause packets with a value of 0 that RO have be transmitted. Only the lower 13 bits of this register are valid.	0		X	X	X	X

EMAC FLOWCONTROLDONE REGISTER (EMAC_TX_STAT_FLOWCONTROLDONE, OFFSET 0x1614)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This register returns the number times that the transmitter initiates flow control by RO transmitting the first XOFF packet. This is slightly different than the outxonsent and outxonsent statistics that count actual pause packets transmitted. Only the lower 12 bits of this register are valid.	0		X	X	X	X

EMAC DOT3STATSSINGLECOLLISIONFRAMES REGISTER (EMAC_TX_STAT_DOT3STATSSINGLECOLLISIONFRAMES, OFFSET 0x1618)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	06	08	09	16
This register returns the number of collisions that were followed by successful packet RO transmits. This is the same as the number of packets that were transmitted with only one collision. Only the lower 13 bits of this register are valid.	0		X	X	X	X

EMAC DOT3STATSMULTIPLECOLLISIONFRAMES REGISTER**(EMAC_TX_STAT_DOT3STATSMULTIPLECOLLISIONFRAMES, OFFSET 0x161c)**

Description	Mode	Reset	06	08	09	16
This register returns the number of packets that have transmitted with more than one RO collision. Only the lower 13 bits of this register are valid.	0		X	X	X	X

EMAC DOT3STATSDEFERREDTRANSMISSIONS REGISTER**(EMAC_TX_STAT_DOT3STATSDEFERREDTRANSMISSIONS, OFFSET 0x1620)**

Description	Mode	Reset	06	08	09	16
This register returns the number of packets that were delayed in transmission because they had to wait for a RX packet to complete. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

**EMAC DOT3STATSEXCESSIVECOLLISIONS REGISTER
(EMAC_TX_STAT_DOT3STATSEXCESSIVECOLLISIONS, OFFSET 0x1624)**

Description	Mode	Reset	06	08	09	16
This register returns the number of packets that have been dropped due to having 16 collisions in a row. Only the lower 9 bits of this register count are valid.	RO	0	X	X	X	X

**EMAC DOT3STATSLATECOLLISIONS REGISTER
(EMAC_TX_STAT_DOT3STATSLATECOLLISIONS, OFFSET 0x1628)**

Description	Mode	Reset	06	08	09	16
This register returns the number of packets that have been dropped due to having a collision received after the 64 byte collision window. Only the lower 12 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHcOUTUNICASTPKTS REGISTER (EMAC_TX_STAT_IFHCOUTUCASTPKTS, OFFSET 0x162c)

Description	Mode	Reset	06	08	09	16
This register returns the number of packets transmitted that have unicast destination addresses. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHcOUTMULTICASTPKTS REGISTER (EMAC_TX_STAT_IFHCOUTMULTICASTPKTS, OFFSET 0x1630)

Description	Mode	Reset	06	08	09	16
This register returns the number of packets transmitted that have multicast destination addresses. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC IFHcOUTBROADCASTPKTS REGISTER (EMAC_TX_STAT_IFHCOUTBROADCASTPKTS, OFFSET 0x1634)

Description	Mode	Reset	06	08	09	16
This register returns the number of packets transmitted that have the broadcast destination address. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

**EMAC ETHERSTATSPKTS64OCTETS REGISTER
(EMAC_TX_STAT_ETHERSTATSPKTS64OCTETS, OFFSET 0x1638)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted of 64 bytes in size. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSPKTS65OCTETSTo127OCTETS REGISTER**(EMAC_TX_STAT_ETHERSTATSPKTS65OCTETSTo127OCTETS, OFFSET 0x163c)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted that are 65 bytes to 127 bytes in size. Only the lower 13 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSPKTS128OCTETSTo255OCTETS REGISTER**(EMAC_TX_STAT_ETHERSTATSPKTS128OCTETSTo255OCTETS, OFFSET 0x1640)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted that are 128 bytes to 255 bytes in size. Only the lower 11 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSPKTS256OCTETSTo511OCTETS REGISTER**(EMAC_TX_STAT_ETHERSTATSPKTS256OCTETSTo511OCTETS, OFFSET 0x1644)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted that are 256 bytes to 511 bytes in size. Only the lower 10 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSPKTS512OCTETSTo1023OCTETS REGISTER**(EMAC_TX_STAT_ETHERSTATSPKTS512OCTETSTo1023OCTETS, OFFSET 0x1648)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted of 512 bytes to 1023 bytes in size. Only the lower 9 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSPKTS1024OCTETSTo1522OCTETS REGISTER**(EMAC_TX_STAT_ETHERSTATSPKTS1024OCTETSTo1522OCTETS, OFFSET 0x164c)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted of 1024 bytes to 1522 bytes in size. Only the lower 9 bits of this register are valid.	RO	0	X	X	X	X

EMAC ETHERSTATSPKTS1522OCTETSTo9022OCTETS REGISTER**(EMAC_TX_STAT_ETHERSTATSPKTS1522OCTETSTo9022OCTETS, OFFSET 0x1650)**

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames transmitted of 1523 bytes to 9022 bytes in size. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X



EMAC DOT3STATSINTERNALMACTRANSMITTERORS REGISTER**(EMAC_TX_STAT_DOT3STATSINTERNALMACTRANSMITTERORS, OFFSET 0x1654)**

Description	Mode	Reset	06	08	09	16
This register returns the number of times that a frame was dropped due to internal problems in the transmit buffering system. Only the lower 13 bits of this register are valid. This needs more detail to help customers troubleshooting problems where this register increments.	RO	0	X	X	X	X

EMAC TRANSMIT STATISTICS AUTO-CLEAR REGISTERS (EMAC_TX_STAT_AC[22], OFFSET 0x1680)

Description	Mode	Reset	06	08	09	16
These are the auto-clear versions of the above TX statistics registers. These locations are used by the Host Coalescing block to aggregate statistics into its local memory.	AC	0	X	X	X	X

EMAC TRANSMIT RATE LIMIT CONTROL REGISTER (EMAC_TX_RATE_LIMIT_CTRL, OFFSET 0x16FC)
Table 204: Transmit Rate Limit Control Register (emac_tx_rate_limit_ctrl, Offset 0x16fc)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RATE_LIMITER_EN	Rate limiter enable bit. When set, this bit enables the rate limiting function at the start of the next packet. The TX throttle increment and numerator must be programmed at the time this bit is set.	RW	0	-	-	X	X
30–23	RESERVED		RO	0	-	-	X	X
22–16	TX_THROTTLE_NUM	TX throttle numerator. This value specifies the proportion of link bandwidth to be used. The effective used link bandwidth is roughly calculated as TX_THROTTLE_NUM / (TX_THROTTLE_NUM + TX_THROTTLE_INC).	RW	0	-	-	X	X
15–7	RESERVED		RO	0	-	-	X	X
6–0	TX_THROTTLE_INC	TX throttle increment. This value specifies the proportion of unused link bandwidth.	RW	0	-	-	X	X

RECEIVE PARSER MAC (RPM) REGISTERS

The purpose of the Receive Parser MAC (RPM) is to parse received data frames and perform a number of checks before passing the frame down the receive path for processing. As the received frame data flows into the RPM, frame cracking (VLAN, IPv4, TCP, and UDP header parsing), checksum calculation, consistency checks, and rule checking functions are performed on the data. The data is also temporarily buffered in a data FIFO and then moved into RX Mbuf (RBUF) for storage. Once the entire frame is received, if the frame is good, then parsing results and RBUF storage information are placed into the RX Lookup FTQ (RLUPQ). If the frame is bad, then parsing results are dropped and RBUF storage space is returned to the free pool.

RPM COMMAND REGISTER (RPM_COMMAND, OFFSET 0x1800)

Table 205: RPM Command Register (rpm_command, Offset 0x1800)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–5	RESERVED		RO	0	X	X	X	X
4	OVERRUN_ABORT	This bit is set each time the RX FIFO over-runs as an error condition. This means that the action was unrecoverable for some reason. Normally, a packet should be dropped before this overrun occurs.	WC	0	X	X	X	X
3–1	RESERVED		RO	0	X	X	X	X
0	ENABLED	This bit indicates the current enable status of the RPM block. If this bit is 1, it indicates that the block is enabled from the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232.	RO	0	X	X	X	X

RPM STATUS REGISTER (RPM_STATUS, OFFSET 0x1804)

Table 206: RPM Status Register (rpm_status, Offset 0x1804)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–2	RESERVED		RO	0	X	X	X	X
1	FREE_WAIT	This bit indicates that the RX Parser is waiting for a free of an MBUF cluster for a packet that was dropped.	RO	0	X	X	X	X
0	MBUF_WAIT	This bit indicates that the RX Parser is waiting on MBUF space to become available for storing a packet that is being received.	RO	0	X	X	X	X

RPM CONFIGURATION REGISTER (RPM_CONFIG, OFFSET 0x1808)*Table 207: RPM Configuration Register (rpm_config, Offset 0x1808)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	IGNORE_VLAN	When this bit is set, the Frame cracker results and Rule Checker programming would be identical for the same packet received with or without a VLAN tag. When this bit is clear, the VLAN tag will be a part of the Offsets and must be compensated for in programming the Rule Checker.	RW	0x1	X	X	X	X
30	RESERVED		RO	0x0	X	X	-	-
30	DISABLE_WOL_ASSERT	When this bit is set the hardware will not assert the WOL detect signal that generates PME output. This bit does not disable the interesting packet compare feature itself and the result for the compare will be presented to the processors downstream so they can act accordingly.	RW	0x0	-	-	X	X
29–8	RESERVED		RO	0x0	X	X	X	X
7–4	SORT_VECT_VAL	In the Receive Parser Catch-Up (RPC) block, this field controls the sort vector value that all catch-up packets are presented to the RX Processor with. Bits 3–1 of this field have no effect. Bit 0 of this field enables the ACPI interesting packet compare feature to generate PME output. Normally this is left at 1, but can be set to 0 to disable PME output without disabling the entire interesting packet compare feature. In NetXtreme IIA0, bit 0 has no effect.	RW	0x1	X	X	X	X
3	MP_KEEP	When this bit is set, Magic packets will be kept in the RX Data stream. If this bit is cleared, Magic packets are automatically dropped when detected. Magic Packet detection is enabled in the "EMAC Mode Register (emac_mode, Offset 0x1400)" on page 312 .	RW	0x0	X	X	X	X
2	ACPI_KEEP	When this bit is set, ACPI packets will be kept in the RX Data stream. If this bit is cleared, ACPI packets are automatically dropped when detected.	RW	0x0	X	X	X	X
1	ACPI_ENA	When this bit is set, ACPI packet recognition is enabled. This bit must not be enabled until after the pattern has been programmed into the TX Header Buffer and the EMAC, RX Parser, and TX Header Buffer blocks have been enabled via the MISC block.	RW	0x0	X	X	X	X

Table 207: RPM Configuration Register (rpm_config, Offset 0x1808) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	NO_PSD_HDR_CKSUM	This bit is passed to the Checksum section of the parsing block for use in determining the operation of the TCP and UDP checksum.	RW	0x0	X	X	X	X

RPM MANAGEMENT PACKET CONTROL REGISTER (RPM_MGMT_PKT_CTRL, OFFSET 0x180C)*Table 208: RPM Management Packet Control Register (rpm_mgmt_pkt_ctrl, offset 0x180c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–30	RESERVED		RO	0	X	X	–	–
31	MGMT_EN	When this bit is set PRM will detect management packets and inform the MCP by writing a record into MCPQ instead of the RLUPQ. When this bit is clear RPM does not identify management packets and writes RLUPQ records as normal.	RW	0	–	–	X	X
30	MGMT_DISCARD_EN	This field is only valid when the MGMT_EN field is set. When this bit is set RPM will discard all non-management packets. When this bit is clear all packets that pass the acceptance criteria are kept and the MCP performs the management packet detection and discard of non-management packets.	RW	0x1	–	–	X	X
29–8	RESERVED		RO	0	X	X	X	X
7–0	RESERVED		RO	0	X	X	–	–
7–4	MGMT_RULE	This field is a bitmap that identifies the RULE_Px bit in the RLUPQ bits status associated with the management entity. MGMT_RULE[3] corresponds to RULE_P4, MGMT_RULE[2] corresponds to RULE_P3, etc. By default RULE_P4 is associated with the management entity.	RW	0x8	–	–	X	X
3–0	MGMT_SORT	This field is a bitmap that identifies the management entity as a user. By default the management entity is User 3.	RW	0x8	–	–	X	X

RPM VLAN MATCH 0 REGISTER (RPM_VLAN_MATCH0, OFFSET 0x1810)*Table 209: RPM VLAN Match 0 Register (rpm_vlan_match0, Offset 0x1810)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–12	RESERVED		RO	0x0	X	X	X	X

Table 209: RPM VLAN Match 0 Register (rpm_vlan_match0, Offset 0x1810) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11-0	RPM_VLAN_MTCH0_VALUE	This value will match to the bottom 12 bits of the VLAN tag of each packet (if it has one). If a match is found, bit 0 of the STAT_VLAN_MATCH field in the RX Lookup FTQ entry for the packet will be set. This value is ignored if the ENABLE flag is not set.	RW	0x0	X	X	X	X

RPM VLAN MATCH 1 REGISTER (RPM_VLAN_MATCH1, OFFSET 0x1814)**Table 210: RPM VLAN Match 1 Register (rpm_vlan_match1, Offset 0x1814)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31-12	RESERVED		RO	0x0	X	X	X	X
11-0	RPM_VLAN_MTCH1_VALUE	This value will match to the bottom 12 bits of the VLAN tag of each packet (if it has one). If a match is found, bit 1 of the STAT_VLAN_MATCH field in the RX Lookup FTQ entry for the packet will be set. This value is ignored if the ENABLE flag is not set.	RW	0x0	X	X	X	X

RPM VLAN MATCH 2 REGISTER (RPM_VLAN_MATCH2, OFFSET 0x1818)**Table 211: RPM VLAN Match 2 Register (rpm_vlan_match2, Offset 0x1818)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31-12	RESERVED		RO	0x0	X	X	X	X
11-0	RPM_VLAN_MTCH2_VALUE	This value will match to the bottom 12 bits of the VLAN tag of each packet (if it has one). If a match is found, bit 2 of the STAT_VLAN_MATCH field in the RX Lookup FTQ entry for the packet will be set. This value is ignored if the ENABLE flag is not set.	RW	0x0	X	X	X	X

RPM VLAN MATCH 3 REGISTER (RPM_VLAN_MATCH3, OFFSET 0x181C)**Table 212: RPM VLAN Match 3 Register (rpm_vlan_match3, Offset 0x181c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31-13	RESERVED		RO	0x0	X	X	X	X
11-0	RPM_VLAN_MTCH3_VALUE	This value will match to the bottom 12 bits of the VLAN tag of each packet (if it has one). If a match is found, bit 3 of the STAT_VLAN_MATCH field in the RX Lookup FTQ entry for the packet will be set. This value is ignored if the ENABLE flag is not set.	RW	0x0	X	X	X	X

RPM SORT USER 0 REGISTER (RPM_SORT_USER0, OFFSET 0x1820)*Table 213: RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	ENA	When this is 1, it enables packets for user 0. When this bit is 0, then no packets will be delivered to user 0. Other bits in this register must only be changed when this bit is 0: to modify this register, this bit must first be written as 0 with the original register value, followed by a 2nd write to reconfigure, followed by a third write with the new value and this bit as a 1.	RW	0	X	X	X	X
30–26	RESERVED		RO	0	X	X	X	X
25	RESERVED		RO	0	X	X	—	—
25	VLAN_NOTMATCH	When this bit is set it indicates that the VLAN hit signal will be indicated for any packet that has the VLAN tag.	RW	0	—	—	X	X
24	PROM_VLAN	When this is 1, it indicates that user 0 does not care what VLAN its packets are on and that VLAN should be ignored. If this bit is 0, then one or more of the VLAN_EN bits should be set to allow packets to be routed to user 0.	RW	0	X	X	X	X
23–20	VLAN_EN	Each bit in this field enables the corresponding value in the VLAN filter to be allowed for user 0. If any of these bits are set, then the setting of PROM_VLAN has no effect for user 0 and only packets that are on the selected VLAN values will be enabled for user 0.	RW	0	X	X	X	X
19	PROM_EN	This bit enables all received packets to be delivered to user 0. If any user selects this option, then no packets will be dropped.	RW	0	X	X	X	X
18	MC_HSH_EN	This bit enables packets with the multicast addresses that hit in the multicast hash table to be delivered to user 0.	RW	0	X	X	X	X
17	MC_EN	This bit enables packets with multicast addresses to be delivered to user 0. If any user has selected to receive all Multicast packets with this bit, then no multicast packets will be dropped or counted in the filter_discards statistic. If no user has selected to receive all Multicast packets, then multicast packet dropping is controlled by the multicast hash table.	RW	0	X	X	X	X

Table 213: RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
16	BC_EN	This bit enables packets with the broadcast address to be delivered to user 0. If the SORT_MODE field (see "EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8)" on page 327) is 1, then no user has enabled delivery of broadcast packets, broadcast packets will be dropped and counted in the filter_discards statistic.	RW	0	X	X	X	X
15–0	PM_EN	Each bit in this field enables the corresponding value in the perfect match filter of the EMAC block for user 0. All unicast packets that match one of the perfect match filters enabled by user 0 will be delivered to user 0. If the SORT_MODE field (see "EMAC RX Mode Register (emac_rx_mode, Offset 0x14c8)" on page 327) is 1, then any unicast packet that is not matched by any of the users will be dropped and counted in the filter_discards statistic.	RW	0	X	X	X	X

RPM SORT USER 1 REGISTER (RPM_SORT_USER1, OFFSET 0x1824)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820)" on page 345.	RO	0	X	X	X	X

RPM SORT USER 2 REGISTER (RPM_SORT_USER2, OFFSET 0x1828)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820)" on page 345.	RO	0	X	X	X	X

RPM SORT USER 3 REGISTER (RPM_SORT_USER3, OFFSET 0x182C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Sort User 0 Register (rpm_sort_user0, Offset 0x1820)" on page 345.	RO	0	X	X	X	X

RPM L2 FILTER DISCARDS STATISTIC REGISTER (RPM_STAT_L2_FILTER_DISCARDS, OFFSET 0x1840)

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames dropped due the layer 2 perfect match, broadcast, multicast filters, and MAC control frame drops. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X

RPM RULE CHECK DISCARDS STATISTIC REGISTER (RPM_STAT_RULE_CHECKER_DISCARDS, OFFSET 0x1844)

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames dropped due rule the checker. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X

RPM IFINFTQDISCARDS STATISTIC REGISTER (RPM_STAT_IFINFTQDISCARDS, OFFSET 0x1848)

Description	Mode	Reset	06	08	09	16
This register returns the number of packets dropped because the FTQ to the RX Lookup block was full. This statistic will only count if there is sufficient storage in the RX MBUF area for the packet. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X

RPM IFINMBUFDISCARD STATISTIC REGISTER (RPM_STAT_IFINMBUFDISCARD, OFFSET 0x184C)

Description	Mode	Reset	06	08	09	16
This register returns the number of packets dropped because the RPM was unable to get sufficient MBUF blocks to store the packet. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X

RPM RULE CHECKER P4 HITS STATISTIC REGISTER (RPM_STAT_RULE_CHECKER_P4_HIT, OFFSET 0x1850)

Description	Mode	Reset	06	08	09	16
This register returns the number of good frames that have activated each rule in the rule checker. Only the lower 17 bits of this register are valid.	RO	0	X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 0 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION0, OFFSET 0x1854)

Table 214: RPM IPv6 Programmable Extension 0 Register (rpm_ipv6_programmable_extension0, offset 0x1854)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	RESERVED		RO	0	X	X	-	-
31	NEXT_HEADER_EN	When this bit is set it enables parsing of programmable extension header 0.	RW	0	-	-	X	X
30	NEXT_HEADER_LEN_TYPE	When this bit is set the controller will use the header length specified in the NEXT_HEADER_LEN field and not the value specified in the packet.	RW	0	-	-	X	X
29-24	RESERVED		RO	0	-	-	X	X
23-16	NEXT_HEADER	This field is the programmable extension header selector.	RW	0	-	-	X	X
15-8	RESERVED		RO	0	-	-	X	X
7-0	NEXT_HEADER_LEN	This field is the extension header length and is only valid when the NEXT_HEADER_LEN_TYPE bit is set. The length is given in 8 byte units.	RW	0	-	-	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 1 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION1, OFFSET 0x1858)

Description	Mode	Reset	06	08	09	16
See the definition for "RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)" on page 348.	RO	0	X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 2 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION2, OFFSET 0x185c)

Description	Mode	Reset	06	08	09	16
See the definition for "RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)" on page 348.	RO	0	X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 3 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION3, OFFSET 0x1860)

Description	Mode	Reset	06	08	09	16
See the definition for "RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)" on page 348.	RO	0	X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 4 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION4, OFFSET 0x1864)

Description	Mode	Reset	06	08	09	16
See the definition for “ RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)” on page 348.	0		X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 5 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION5, OFFSET 0x1868)

Description	Mode	Reset	06	08	09	16
See the definition for “ RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)” on page 348.	0		X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 6 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION6, OFFSET 0x186C)

Description	Mode	Reset	06	08	09	16
See the definition for “ RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)” on page 348.	0		X	X	X	X

RPM IPv6 PROGRAMMABLE EXTENSION 7 REGISTER (RPM_IPV6_PROGRAMMABLE_EXTENSION7, OFFSET 0x1870)

Description	Mode	Reset	06	08	09	16
See the definition for “ RPM IPv6 Programmable Extension 0 Register RO (rpm_ipv6_programmable_extension0, offset 0x1854)” on page 348.	0		X	X	X	X

RPM STATISTICS AUTO-CLEAR REGISTERS (RPM_STAT_AC[5], OFFSET 0x1880)

Description	Mode	Reset	06	08	09	16
These are the auto-clear versions of the above statistics registers. These locations are AC used by the Host Coalescing block to aggregate statistics into its local memory.	0		X	X	X	X

RPM RULE CHECK CONTROL 16 REGISTER (RPM_RC_CNTL_16, OFFSET 0x18E0)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the “ RPM Rule Check Control 0 Register RO (rpm_rc_ctrl_0, Offset 0x1900)” on page 351.	0		-	-	X	X

RPM RULE CHECK VALUE/MASK 16 REGISTER (RPM_RC_VALUE_MASK_16, OFFSET 0x18E4)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0	-	-	X	X	

RPM RULE CHECK CONTROL 17 REGISTER (RPM_RC_CNTL_17, OFFSET 0x18E8)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	-	-	X	X	

RPM RULE CHECK VALUE/MASK 17 REGISTER (RPM_RC_VALUE_MASK_17, OFFSET 0x18EC)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0	-	-	X	X	

RPM RULE CHECK CONTROL 18 REGISTER (RPM_RC_CNTL_18, OFFSET 0x18F0)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	-	-	X	X	

RPM RULE CHECK VALUE/MASK 18 REGISTER (RPM_RC_VALUE_MASK_18, OFFSET 0x18F4)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0	-	-	X	X	

RPM RULE CHECK CONTROL 19 REGISTER (RPM_RC_CNTL_19, OFFSET 0x18F8)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	-	-	X	X	

RPM RULE CHECK VALUE/MASK 19 REGISTER (RPM_RC_VALUE_MASK_19, OFFSET 0x18FC)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	RO	0	-	-	X	X

RPM RULE CHECK CONTROL 0 REGISTER (RPM_RC_CNTL_0, OFFSET 0x1900)

Table 215: RPM Rule Check Control 0 Register (rpm_rc_CNTL_0, Offset 0x1900)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30	NBIT	This bit is combined with the SBIT value to determine how this rule is combined with other rules. When this bit is set, this rule and the next rule must both match for an action to happen.	RW	0	X	X	X	X
29	P3	This field is used to pass rule indications to the RXP or COM blocks. The value of these bits is defined by firmware. The logical OR of these bits from all the rules that match a packet are placed in the status record for that packet.	RW	0	X	X	X	X
28	P2	This field is used to pass rule indications to the RXP or COM blocks. The value of these bits is defined by firmware. The logical OR of these bits from all the rules that match a packet are placed in the status record for that packet.	RW	0	X	X	X	X
27	P1	This field is used to pass rule indications to the RXP or COM blocks. The value of these bits is defined by firmware. The logical OR of these bits from all the rules that match a packet are placed in the status record for that packet.	RW	0	X	X	X	X
26	MASK	This bit indicates that the corresponding value mask register is to be treated as 16 bits of value and 16 bits of mask instead of 32 bits of value.	RW	0	X	X	X	X
25	DISCARD	When this bit is set, frames matching this rule will be discarded.	RW	0	X	X	X	X
24	MAP	This bit indicates that the masked value should be passed as the "class" field on output to the RX Processor. The MASK mode must be used and the first least significant unmasked bit and the 2 bits more significant than that one in the field will be used as the class value.	RW	0	X	X	X	X
23–20	CMDSEL	This field is used when combining two rules groups using the SBIT.	RW	0	X	X	X	X

Table 215: RPM Rule Check Control 0 Register (rpm_rc_ctrl_0, Offset 0x1900) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
19	SBIT	This bit is combined with the NBIT value to define how this rule is combined with other rules. What does it mean when this bit is set?	RW	0	X	X	X	X
18	RESERVED		RO	0	X	X	-	-
18	MAP	This bit indicates that the masked value should be passed as the "class" field on output to the RX processor. The MASK field must be used and the first, least significant, un-masked bit, along with the 2 more significant bits than that one in the field, will be used as the class value.	RW	0	-	-	X	X
17–16	COMP	This field selects the comparison operator that will be used for this rule.	RW	0	X	X	X	X
Value Name								
	0 EQUAL	The rule will be true if the frame value is equal to the value_mask register.						
	1 NEQUAL	The rule will be true if the frame value is not equal to the value_mask register.						
	2 GREATER	The rule will be true if the frame value is greater than the value_mask register.						
	3 LESS	The rule will be true if the frame value is less than the value_mask register.						

Table 215: RPM Rule Check Control 0 Register (rpm_rc_ctrl_0, Offset 0x1900) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–13	HDR_TYPE	This field selects the header within which the comparison to the “RPM Rule Check Value/ Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)” on page 354 will be done. If the specified starting point in the packet is not available in the frame, then the rule will be false for that frame.	RW	0	X	X	X	X
		Value Name						
		0 START	The Offset will be from the start of the frame.					
		1 IP	The Offset will be from the start of the IP header.					
		2 TCP	The Offset will be from the start of the TCP header.					
		3 UDP	The Offset will be from the start of the UDP header.					
		4 DATA	The Offset will be from the start of the payload of the frame.					
		5 TCP_UDP	The offset will be from the start of the TCP or UDP header. (BCM5709 and BCM5716 only)					
		6 ICMPV6	IPv6 only. The offset will be from the start of the ICMPv6 header. (BCM5709 and BCM5716 only)					
12	P4	This field is used to pass rule indications to the RXP or COM blocks. The value of these bits is defined by firmware. The logical OR of these bits from all the rules that match a packet are placed in the status record for that packet.	RW	0	X	X	X	X
11	PRIORITY	The logical OR of this bit in all the rules that match the packet is placed in the status record for each packet.	RW	0	X	X	X	X
10–8	CLASS	This field is passed to the RXP and COM blocks for use in classifying the incoming traffic. The lowest numeric CLASS value of all the matching rules is what is output in the status record. If no rules match, then the CLASS field of the “RPM Rule Check Configuration Register (rpm_rc_config, Offset 0x1980)” on page 359 is passed.	RW	0	X	X	X	X
7–0	OFFSET	This field defines the byte Offset into the packet from the specified HD_TYP to start comparison.	RW	0	X	X	X	X

RPM RULE CHECK VALUE/MASK 0 REGISTER (RPM_RC_VALUE_MASK_0, OFFSET 0x1904)*Table 216: RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	MASK	When the MASK field of the “RPM Rule Check Control 0 Register (rpm_rc_ctrl_0, Offset 0x1900)” on page 351 is set to 1, this value is used to mask the comparison of the value. A 1 in this area causes the corresponding bit in the VALUE area to be unchanged.	RW	0	X	X	X	X
15–0	VALUE	When the MASK field of the “RPM Rule Check Control 0 Register (rpm_rc_ctrl_0, Offset 0x1900)” on page 351 is set to 1, this value is used as the comparison value at the specified Offset.	RW	0	X	X	X	X

RPM RULE CHECK CONTROL 1 REGISTER (RPM_RC_CNTL_1, OFFSET 0x1908)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the “RPM Rule Check Control 0 Register (rpm_rc_ctrl_0, Offset 0x1900)” on page 351.	RO	0	X	X	X	X

RPM RULE CHECK VALUE/MASK 1 REGISTER (RPM_RC_VALUE_MASK_1, OFFSET 0x190c)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the “RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)” on page 354.	RO	0	X	X	X	X

RPM RULE CHECK CONTROL 2 REGISTER (RPM_RC_CNTL_2, OFFSET 0x1910)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the “RPM Rule Check Control 0 Register (rpm_rc_ctrl_0, Offset 0x1900)” on page 351.	RO	0	X	X	X	X

RPM RULE CHECK VALUE/MASK 2 REGISTER (RPM_RC_VALUE_MASK_2, OFFSET 0x1914)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the “RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)” on page 354.	RO	0	X	X	X	X

RPM RULE CHECK CONTROL 3 REGISTER (RPM_RC_CNTL_3, OFFSET 0x1918)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	X X X X				

RPM RULE CHECK VALUE/MASK 3 REGISTER (RPM_RC_VALUE_MASK_3, OFFSET 0x191C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0	X X X X				

RPM RULE CHECK CONTROL 4 REGISTER (RPM_RC_CNTL_4, OFFSET 0x1920)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	X X X X				

RPM RULE CHECK VALUE/MASK 4 REGISTER (RPM_RC_VALUE_MASK_4, OFFSET 0x1924)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0	X X X X				

RPM RULE CHECK CONTROL 5 REGISTER (RPM_RC_CNTL_5, OFFSET 0x1928)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	X X X X				

RPM RULE CHECK VALUE/MASK 5 REGISTER (RPM_RC_VALUE_MASK_5, OFFSET 0x192C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0	X X X X				

RPM RULE CHECK CONTROL 6 REGISTER (RPM_RC_CNTL_6, OFFSET 0x1930)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0	X X X X				

RPM RULE CHECK VALUE/MASK 6 REGISTER (RPM_RC_VALUE_MASK_6, OFFSET 0x1934)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 7 REGISTER (RPM_RC_CNTL_7, OFFSET 0x1938)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 7 REGISTER (RPM_RC_VALUE_MASK_7, OFFSET 0x193C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 8 REGISTER (RPM_RC_CNTL_8, OFFSET 0x1940)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 8 REGISTER (RPM_RC_VALUE_MASK_8, OFFSET 0x1944)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 9 REGISTER (RPM_RC_CNTL_9, OFFSET 0x1948)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 9 REGISTER (RPM_RC_VALUE_MASK_9, OFFSET 0x194C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 10 REGISTER (RPM_RC_CNTL_10, OFFSET 0x1950)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 10 REGISTER (RPM_RC_VALUE_MASK_10, OFFSET 0x1954)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 11 REGISTER (RPM_RC_CNTL_11, OFFSET 0x1958)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 11 REGISTER (RPM_RC_VALUE_MASK_11, OFFSET 0x195C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 RO Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 12 REGISTER (RPM_RC_CNTL_12, OFFSET 0x1960)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 RO Register (rpm_rc_ctrl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 12 REGISTER (RPM_RC_VALUE_MASK_12, OFFSET 0x1964)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 13 REGISTER (RPM_RC_CNTL_13, OFFSET 0x1968)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 Register (rpm_rc_cntl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 13 REGISTER (RPM_RC_VALUE_MASK_13, OFFSET 0x196C)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 14 REGISTER (RPM_RC_CNTL_14, OFFSET 0x1970)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 Register (rpm_rc_cntl_0, Offset 0x1900)" on page 351.	0		X	X	X	X

RPM RULE CHECK VALUE/MASK 14 REGISTER (RPM_RC_VALUE_MASK_14, OFFSET 0x1974)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	0		X	X	X	X

RPM RULE CHECK CONTROL 15 REGISTER (RPM_RC_CNTL_15, OFFSET 0x1978)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Control 0 Register (rpm_rc_cntl_0, Offset 0x1900)" on page 351.	0		X	X	X	X



RPM RULE CHECK VALUE/MASK 15 REGISTER (RPM_RC_VALUE_MASK_15, OFFSET 0x197c)

Description	Mode	Reset	06	08	09	16
This register has the same bit field definitions as the "RPM Rule Check Value/Mask 0 Register (rpm_rc_value_mask_0, Offset 0x1904)" on page 354.	RO	0	X	X	X	X

RPM RULE CHECK CONFIGURATION REGISTER (RPM_RC_CONFIG, OFFSET 0x1980)

Table 217: RPM Rule Check Configuration Register (rpm_rc_config, Offset 0x1980)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	-	-
31	KNUM_OVERWRITE	When this bit is set the RPM will overwrite the upper 4 bits of the KNUM field in the RLUPQ record with rule match bits. Match bits for rules 16 to 19 are mapped to bits 4 to 7 of the KNUM field. The lower 4 bits of the KNUM field retain the original functionality where they are incremented for every received frame.			-	-	X	X
30–27	RESERVED				X	X	X	X
26–24	DEF_CLASS	This field defines the default class that will be passed on all frames that do not match any rules.	RW	0x7	X	X	X	X
23–16	RESERVED		RO	0	X	X	-	-
23–20	RESERVED		RO	0	-	-	X	X
19–0	RULE_ENABLE	This field provides a bit-per-rule enable capability. When the bit is set, the rule is enabled. When cleared, the rule will not be found. This bit may be changed at any time.			-	-	X	X
15–0	RULE_ENABLE	This field provides a bit-per-rule enable capability. When the bit is set, the rule is enabled. When cleared, the rule will not be found. This bit may be changed at any time.	RW	0	X	X	-	-

The purpose of the Receive Lookup (RLUP) block is to identify off-loaded IPv4/IPv6 connections by comparing the 4-tuple extracted from the received packet to the values stored in the on-chip lookup table. The 4-tuple itself consists of the source and destination IP addresses and the source and destination TCP ports. In order to identify open TCP port the RPLUP also performs a 2-tuple lookup which consists of the destination IP address and the destination TCP port. The RLUP also calculates a 32-bit RSS hash value for both the IPv4 and IPv6 connections which is performed on a 2-tuple or 4-tuple according to RLUP register settings. The results of the RLUP processing are passed to the Receive Processor (RXP) using the RXP FTQ.

RLUP COMMAND REGISTER (RLUP_COMMAND, OFFSET 0x2000)

Table 218: RLUP Command Register (rlup_command, Offset 0x2000)

Bit	Name	Description	Mode	Reset	06	08	09	16															
31–13	RESERVED		RO	0	X	X	X	X															
12–7	RESERVED		RO	0	X	X	–	–															
12	MAINTENANCE_MODE	When this bit is set the RLUP only services operations specified in bits [6:1] of this register and ignores input from the RLUP FTQ. All side-effects of setting this bit are the responsibility of software.	RW	0	–	–	X	X															
11	2ND_TUPLE_LOOKUP_EN	When this bit is set 2-tuple lookup is also performed for all packets which 4-tuple lookup is performed.	RW	0	–	–	X	X															
10–9	ENTRY_TYPE	This field qualifies operations specified in bits [5:1] of this register. In the case of 2-tuple write operations software needs to insure that the source IP address and the source TCP port are set to 0. 0 = IPv4, 1 = IPv6_4_TUPLE, 2 = IPv6_2_TUPLE, 3 = RES	RW	0	–	–	X	X															
		<table border="1"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>IPv4</td> <td>–</td> </tr> <tr> <td>1</td> <td>IPv6_4_TU PLE</td> <td>–</td> </tr> <tr> <td>2</td> <td>IPv6_2_TU PLE</td> <td>–</td> </tr> <tr> <td>3</td> <td>RES</td> <td>–</td> </tr> </tbody> </table>	Value	Name	Description	0	IPv4	–	1	IPv6_4_TU PLE	–	2	IPv6_2_TU PLE	–	3	RES	–						
Value	Name	Description																					
0	IPv4	–																					
1	IPv6_4_TU PLE	–																					
2	IPv6_2_TU PLE	–																					
3	RES	–																					
8	WRITE_RAM	Setting this bit writes the CAM location defined by the idx register. Data is stored in the CID register bits [11:0]. This bit should not be written unless it reads as 0 first.	SC	0	–	–	X	X															
7	READ_RAM	Setting this bit reads the CAM location defined by the idx register. Data is provided by the CID register bits [11:0]. This bit should not be written unless it reads as 0 first.	SC	0	–	–	X	X															
6	CAM_RESET	This bit is a self-clearing command to clear the current content of the CAM. All entries will be invalidated. This bit clears when each operation is finished. This bit should not be written as 1 unless it reads as a 0 before starting.	SC	0	X	X	X	X															

Table 218: RLUP Command Register (rlup_command, Offset 0x2000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
5	WRITE	This bit is a self-clearing command to write the CAM location as defined by the "RLUP Index Register (rlup_idx, Offset 0x2018)" on page 367 . The Tuple, CID and VALID values must be programmed in the appropriate registers (what are the appropriate registers?) to write correct data into the CAM location. This bit clears when each operation is finished. This bit should not be written as 1 unless it reads as a 0 before starting.	SC	0	X	X	X	X
4	READ	This bit is a self-clearing command to read the CAM location as defined by the "RLUP Index Register (rlup_idx, Offset 0x2018)" on page 367 . The Tuple, CID and VALID bit values will be updated at the end of the operation. This bit clears when each operation is finished. This bit should not be written as 1 unless it reads as a 0 before starting.	SC	0	X	X	X	X
3	LOOKUP	This bit is a self-clearing command to do a manual lookup of the current 4-Tuple value from the tuple registers into the CAM. For the tuple to be found, the VALID bit in the CID must be set. The result (cid valid and valid bit) is in the "RLUP Context ID Register (rlup_cid, Offset 0x2014)" on page 365 . The SUCCESS bit also indicates that the value was found and the VALID was set for the found entry. This bit should not be written as 1 unless it reads as a 0 before starting.	SC	0	X	X	X	X
2	INVALIDATE	This bit is a self-clearing command to clear the current 4-Tuple value from the tuple registers from the CAM. This bit clears when the operation is finished. For the entry to be found, the VALID bit in the CID register must be set. If the operation completes and the SUCCESS bit is set below, then the value was found and removed. This bit should not be written as 1 unless it reads as a 0 before starting.	SC	0	X	X	X	X

Table 218: RLUP Command Register (*rlup_command*, Offset 0x2000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	ADD	This bit is a self-clearing command to add the current 4-Tuple value from the tuple registers into the CAM with the specified CID and VALID bit (below). The value will be added in the first available location in the CAM. If the operation completes and the SUCCESS bit is set below, then the value was added. This bit should not be written as 1 unless it reads as a 0 before starting.	SC	0	X	X	X	X
0	ENABLED	This bit indicates the current enable status of the RLUP block. If this bit is 1, it indicates that the block is enabled from the " "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232.	RO	0	X	X	X	X

RLUP STATUS REGISTER (RPLU_STATUS, OFFSET 0x2004)*Table 219: RLUP Status Register (rplu_status, Offset 0x2004)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–24	ACK_SM	Enumeration:	RO	0	X	X	—	—
		Value Name	Description					
		0 IDLE	idle state					
		1 WAIT	wait for cam access to finish					
		2 STROBE	ack strobe generation state					
25–7	RESERVED		RO	0	—	—	X	X
23–22	RESERVED		RO	0	X	X	—	—
21–20	REGCAM_SM	Enumeration:	RO	0	X	X	—	—
		Value Name	Description					
		0 IDLE	idle state					
		1 STROBE	cam access strobe generation state					
		2 WAIT	wait for cam access to finish					
19	RESERVED		RO	0	X	X	—	—
18–16	LOOKUP_SM	Enumeration:	RO	0	X	X	—	—
		Value Name	Description					
		0 IDLE	idle state					
		1 INPUT	input FTQ entry read					
		2 CAM_GRC	wait for cam access from GRC to finish					
		3 CAM_STROBE	cam lookup strobe generation state					
		4 CAM_WAIT	wait for cam lookup to finish					
		5 FTQ_WR	rx processor FTQ write state					
		6 FTQ_POP	input FTQ entry pop state					
15–4	RESERVED		RO	0	X	X	—	—
6	ZERO_CNT_ERR	This bit is set on IPv6 DEL operation when on RAM first or second IPv6 slice has a leaf count of 0. This should never happen under normal conditions and indicates an inconsistency within the lookup table. The address of the slice that generated this error is stored in IPv6_IDX1 register. This flag should be updated every time a new operation specified in bits [5:1] of the “RLUP Command Register (rlup_command, Offset 0x2000)” on page 360 is executed.	RO	0	—	—	X	X

Table 219: RLUP Status Register (rplu_status, Offset 0x2004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
5	DUPLICATE_ENTRY	This bit is set on an ADD operation whenever an entry already exists within the CAM. This flag will be updated any time a new operation specified in bits [5:1] of "RLUP Command Register (rlup_command, Offset 0x2000)" on page 360 is executed.	RO	0	-	-	X	X
4	CAM_FULL	This bit is set on an ADD operation whenever an IPv4 or IPv6 cannot be stored. Note that in the case of IPv6 this bit may be set even when there are CAM entries available since an IPv6 4-tuple requires 3 available CAM entries. This flag will be updated any time a new operation specified in bits [5:1] of "RLUP Command Register (rlup_command, Offset 0x2000)" on page 360 is executed.	RO	0	-	-	X	X
3	LOOKUP_MATCH_STAT	This generic statistic bit toggles once each time a TCP packet has been looked up and found in the CAM.	RO	0	X	X	-	-
3	2TUPLE_LOOKUP_MAT CH_STAT	This generic statistic bit toggles once each time a 2-tuple has been looked up and found in the CAM.	RO	0	-	-	X	X
2	WORD_MATCH	Individual CAM word line match bit. This bit is used for CAM testing and should not be used for any functional use. What does this mean? Not used by the host driver?	RO	0	X	X	-	-
2	LOOKUP_MATCH_STAT	This generic statistic bit toggles once each time a TCP packet has been looked up and found in the CAM.	RO	0	-	-	X	X
1	SUCCESS	When this bit is a 1, it indicates that the last ADD, INVALIDATE, or LOOKUP found a location to write, remove, or lookup.	RO	0	X	X	X	X
0	FTQ	This bit indicates that the RLUP block is waiting on the output FTQ to drain before proceeding.	RO	0	X	X	X	X

RLUP IP SOURCE ADDRESS REGISTER (RLUP_IPSRC, OFFSET 0x2008)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

RLUP IP DESTINATION ADDRESS REGISTER (RLUP_IPDEST, OFFSET 0x200c)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

RLUP TCP PORT REGISTER (RLUP_TCPORT, OFFSET 0x2010)

This read-write register is used for the ADD, INVALIDATE, and LOOKUP fields of the “[RLUP Command Register \(rlup_command, Offset 0x2000\)](#)” on page 360. This field is used for the TCP port values. The read value of this register is updated on READ operations.

Table 220: RLUP TCP Port Register (rlup_tcport, Offset 0x2010)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	SRCPORT	TCP Source Port Value	RW	0	X	X	X	X
15–0	DESTPORT	TCP Destination Port Value	RW	0	X	X	X	X

RLUP CONTEXT ID REGISTER (RLUP_CID, OFFSET 0x2014)

Table 221: RLUP CID Register (rlup_cid, Offset 0x2014)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	VALID	This is the VALID flag for a particular CAM entry. The write value of this flag is only used for the WRITE, ADD, INVALIDATE, and LOOKUP commands (see the “ RLUP Command Register (rlup_command, Offset 0x2000) ” on page 360) and is ignored for READ commands. This means that this bit must be set to LOOKUP, ADD, INVALIDATE, or WRITE valid entries. The read value is updated on a READ command.	RW	0	X	X	X	X
30–21	RESERVED	—	RO	0	X	X	X	X
20–7	VALUE	This is the CID value to be associated with the Tuple value for the ADD and WRITE commands. This value is updated on the INVALIDATE, LOOKUP, and READ commands.	RW	0	X	X	X	X
6–2	RESERVED	—	RO	0	X	—	—	—
6–0	RESERVED	—	RO	0	—	X	X	X

Table 221: RLUP CID Register (rlup_cid, Offset 0x2014) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1-0	RSS_MODE_RD_VAL	This value configures the operation of the receive side scaling (RSS) logic. When enabled, the RSS logic calculates a hash over the specified packet fields using the RSS key and returns the results in the pseudo_cksum and raw_cksum FTQ entries to the RXP. If RSS is calculated for a packet, then the RSS bits in the frame status word is set to 1.	RW	0	X	-	-	-
Value Name								
	0x0 OFF	RSS is disabled. The pseudo_cksum and raw_cksum values are passed through from the input FTQ, unmodified.						
	0x1 ALL	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash is calculated over the IP source, IP destination, TCP source port, and TCP destination port. The upper 16 bits of the hash result replaces the RX parser pseudo_cksum value. The lower 16 bits of the hash result replaces the RX parser raw_cksum value.						
	0x2 IP_ONLY	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash is calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash is calculated over the IP source, IP destination, TCP source port, and TCP destination port.						
	0x3 RESERVED	-						

RLUP INDEX REGISTER (RLUP_IDX, OFFSET 0x2018)*Table 222: RLUP Index Register (rlup_idx, Offset 0x2018)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–10	RESERVED		RO	0	X	X	X	X
9–0	IDX_VALUE	The write value of this register is only used for the CAM READ and WRITE operations. This value ranges from 0 to 1023 to select one CAM entry value. The read value of this register is the last value written.	RW	0	X	X	X	X

RLUP CONFIGURATION REGISTER (RLUP_CONFIG, OFFSET 0x201c)

Table 223: RLUP Configuration Register (rlup_config, Offset 0x201c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–4	RESERVED		RO	0	X	X	X	X
3–2	IPv6_RSS_TYPE	This value configures the operation of the RSS logic for IPv6. When enabled, the RSS logic calculates a hash over the specified packet fields using the RSS key and returns the results in the pseudo_cksum and raw_cksum FTQ entries to the RXP. If RSS is calculated for a packet, then the RSS bits in the frame status word is set to 1.	WO	0	–	–	X	X
Value	Name	Description						
0x0	OFF	RSS is disabled. The pseudo_cksum and raw_cksum values are passed through from the input FTQ, unmodified.						
0x1	ALL	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port. The upper 16 bits of the hash result will replace the RX parser pseudo_cksum value. The lower 16 bits of the hash result will replace the RX parser raw_cksum value.						
0x2	IP_ONLY	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port.						
0x3	RESERVED							

Table 223: RLUP Configuration Register (rlup_config, Offset 0x201c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16														
1–0	IPv4_RSS_TYPE	This value configures the operation of the RSS logic for IPv4. When enabled, the RSS logic calculates a hash over the specified packet fields using the RSS key and returns the results in the pseudo_cksum and raw_cksum FTQ entries to the RXP. If RSS is calculated for a packet, then the RSS bits in the frame status word is set to 1.	WO	0	X	X	X	X														
<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0x0</td><td>OFF</td><td>RSS is disabled. The pseudo_cksum and raw_cksum values are passed through from the input FTQ, unmodified.</td></tr> <tr> <td>0x1</td><td>ALL</td><td>RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port. The upper 16 bits of the hash result will replace the RX parser pseudo_cksum value. The lower 16 bits of the hash result will replace the RX parser raw_cksum value.</td></tr> <tr> <td>0x2</td><td>IP_ONLY</td><td>RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port.</td></tr> <tr> <td>0x3</td><td>RESERVED</td><td></td></tr> </tbody> </table>								Value	Name	Description	0x0	OFF	RSS is disabled. The pseudo_cksum and raw_cksum values are passed through from the input FTQ, unmodified.	0x1	ALL	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port. The upper 16 bits of the hash result will replace the RX parser pseudo_cksum value. The lower 16 bits of the hash result will replace the RX parser raw_cksum value.	0x2	IP_ONLY	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port.	0x3	RESERVED	
Value	Name	Description																				
0x0	OFF	RSS is disabled. The pseudo_cksum and raw_cksum values are passed through from the input FTQ, unmodified.																				
0x1	ALL	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port. The upper 16 bits of the hash result will replace the RX parser pseudo_cksum value. The lower 16 bits of the hash result will replace the RX parser raw_cksum value.																				
0x2	IP_ONLY	RSS is enabled for IP and TCP packets. If the packet is IP, but not TCP, then the RSS hash will be calculated over the IP source and IP destination addresses. If the packet is TCP, then the RSS hash will be calculated over the IP source, IP destination, TCP source port, and TCP destination port.																				
0x3	RESERVED																					

RLUP RSS KEY 1 REGISTER (RLUP_RSS_KEY1, OFFSET 0x2020)*Table 224: RLUP RSS Key 1 Register (rlup_rss_key1, Offset 0x2020)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–24	BYTE_0	Byte 0 of the RSS key byte stream.	RW	0	X	X	X	X
23–16	BYTE_1	Byte 1 of the RSS key byte stream.	RW	0	X	X	X	X
15–8	BYTE_2	Byte 2 of the RSS key byte stream.	RW	0	X	X	X	X
7–0	BYTE_3	Byte 3 of the RSS key byte stream.	RW	0	X	X	X	X

RLUP RSS KEY 2 REGISTER (RLUP_RSS_KEY2, OFFSET 0x2024)*Table 225: RLUP RSS Key 2 Register (rlup_rss_key2, Offset 0x2024)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–24	BYTE_4	Byte 4 of the RSS key byte stream.	RW	0	X	X	X	X
‘	BYTE_5	Byte 5 of the RSS key byte stream.	RW	0	X	X	X	X
15–8	BYTE_6	Byte 6 of the RSS key byte stream.	RW	0	X	X	X	X
7–0	BYTE_7	Byte 7 of the RSS key byte stream.	RW	0	X	X	X	X

RLUP RSS KEY 3 REGISTER (RLUP_RSS_KEY3, OFFSET 0x2028)*Table 226: RLUP RSS Key 3 Register (rlup_rss_key3, Offset 0x2028)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–24	BYTE_8	Byte 8 of the RSS key byte stream.	RW	0	X	X	X	X
23–16	BYTE_9	Byte 9 of the RSS key byte stream.	RW	0	X	X	X	X
15–8	BYTE_10	Byte 10 of the RSS key byte stream.	RW	0	X	X	X	X
7–0	BYTE_11	Byte 11 of the RSS key byte stream.	RW	0	X	X	X	X

RLUP RSS KEY 4 REGISTER (RLUP_RSS_KEY4, OFFSET 0x202c)*Table 227: RLUP RSS Key 4 Register (rlup_rss_key4, Offset 0x202c)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–24	BYTE_12	Byte 12 of the RSS key byte stream.	RW	0	X	X	X	X
23–16	BYTE_13	Byte 13 of the RSS key byte stream.	RW	0	X	X	X	X
15–8	BYTE_14	Byte 14 of the RSS key byte stream.	RW	0	X	X	X	X
7–0	BYTE_15	Byte 15 of the RSS key byte stream.	RW	0	X	X	X	X

RLUP IPv6 RSS KEY 5 REGISTER (RLUP_IPV6_RSS_KEY5, OFFSET 0x2030)*Table 228: RLUP IPv6 RSS Key 5 Register (rlup_ipv6_rss_key5, Offset 0x2030)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–0	RESERVED		RO	0	X	X	–	–



Table 228: RLUP IPv6 RSS Key 5 Register (rlup_ipv6_rss_key5, Offset 0x2030) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	BYTE_16	Byte 16 of the RSS key byte stream.	RW	0	—	—	X	X
23–16	BYTE_17	Byte 17 of the RSS key byte stream.	RW	0	—	—	X	X
15–8	BYTE_18	Byte 18 of the RSS key byte stream.	RW	0	—	—	X	X
7–0	BYTE_19	Byte 19 of the RSS key byte stream.	RW	0	—	—	X	X

RLUP IPv6 RSS KEY 6 REGISTER (RLUP_IPV6_RSS_KEY6, OFFSET 0x2034)*Table 229: RLUP IPv6 RSS Key 6 Register (rlup_ipv6_rss_key6, Offset 0x2034)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31–24	BYTE_20	Byte 20 of the RSS key byte stream.	RW	0	—	—	X	X
23–16	BYTE_21	Byte 21 of the RSS key byte stream.	RW	0	—	—	X	X
15–8	BYTE_22	Byte 22 of the RSS key byte stream.	RW	0	—	—	X	X
7–0	BYTE_23	Byte 23 of the RSS key byte stream.	RW	0	—	—	X	X

RLUP IPv6 RSS KEY 7 REGISTER (RLUP_IPV6_RSS_KEY7, OFFSET 0x2038)*Table 230: RLUP IPv6 RSS Key 7 Register (rlup_ipv6_rss_key7, Offset 0x2038)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31–24	BYTE_24	Byte 24 of the RSS key byte stream.	RW	0	—	—	X	X
23–16	BYTE_25	Byte 25 of the RSS key byte stream.	RW	0	—	—	X	X
15–8	BYTE_26	Byte 26 of the RSS key byte stream.	RW	0	—	—	X	X
7–0	BYTE_27	Byte 27 of the RSS key byte stream.	RW	0	—	—	X	X

RLUP IPv6 RSS KEY 8 REGISTER (RLUP_IPV6_RSS_KEY8, OFFSET 0x203C)*Table 231: RLUP IPv6 RSS Key 8 Register (rlup_ipv6_rss_key8, Offset 0x203c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31–24	BYTE_28	Byte 28 of the RSS key byte stream.	RW	0	—	—	X	X
23–16	BYTE_29	Byte 29 of the RSS key byte stream.	RW	0	—	—	X	X
15–8	BYTE_30	Byte 30 of the RSS key byte stream.	RW	0	—	—	X	X
7–0	BYTE_31	Byte 31 of the RSS key byte stream.	RW	0	—	—	X	X

RLUP IPv6 RSS KEY 9 REGISTER (RLUP_IPV6_RSS_KEY9, OFFSET 0x2040)*Table 232: RLUP IPv6 RSS Key 9 Register (rlup_ipv6_rss_key9, Offset 0x2040)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–24	BYTE_32	Byte 32 of the RSS key byte stream.	RW	0	–	–	X	X
23–16	BYTE_33	Byte 33 of the RSS key byte stream.	RW	0	–	–	X	X
15–8	BYTE_34	Byte 34 of the RSS key byte stream.	RW	0	–	–	X	X
7–0	BYTE_35	Byte 35 of the RSS key byte stream.	RW	0	–	–	X	X

RLUP IPv6 RSS KEY 10 REGISTER (RLUP_IPV6_RSS_KEY10, OFFSET 0x2044)*Table 233: RLUP IPv6 RSS Key 10 Register (rlup_ipv6_rss_key10, Offset 0x2044)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31–24	BYTE_36	Byte 36 of the RSS key byte stream.	RW	0	–	–	X	X
23–16	BYTE_37	Byte 37 of the RSS key byte stream.	RW	0	–	–	X	X
15–8	BYTE_38	Byte 38 of the RSS key byte stream.	RW	0	–	–	X	X
7–0	BYTE_39	Byte 39 of the RSS key byte stream.	RW	0	–	–	X	X

RLUP RSS COMMAND REGISTER (RLUP_RSS_COMMAND, OFFSET 0x2048)*Table 234: RLUP RSS Command Register (rlup_rss_command, offset 0x2048)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–17	RESERVED		RO	0	X	X	X	X
16–0	RESERVED		RO	0	X	X	–	–
16–14	HASH_MASK	This field specifies how many LSB bits of the calculated hash value are used as address for the hash indirection table. Valid values are 1-7 inclusive. Because the indirection table address is 7 bits wide, if the value of this field is less than 7, then 7 - HASH_MASK upper bits of the address should be set to 0.	RW	0	–	–	X	X
13	READ	When this bit is set the hardware will perform a 32-bit read from the RSS indirection table using the address provided in the RSS_IND_TABLE_ADDR field. The result is stored in the RSS data register. This bit is cleared by hardware when the operation is completed. Software should insure that this bit is 0 before writing to it.	SC	0	–	–	X	X

Table 234: RLUP RSS Command Register (*rlup_rss_command*, offset 0x2048) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	WRITE	When this bit is set the hardware will perform a 32-bit read-modify write into the RSS indirection table using information provided in the RSS_WRITE_MASK and the “RLUP RSS Data Register (<i>rlup_rss_data</i> , offset 0x204c)” on page XXX. This bit is cleared by hardware when the operation is completed. Software should insure that this bit is 0 before writing to it.	SC	0	-	-	X	X
11–4	RSS_WRITE_MASK	Setting any bit of this field will enable writes of the corresponding receive queue ID into the indirection table on the address specified using the RSS_IND_TABLE_ADDR field. Setting bit 11 enables write of bits [31:28], setting bit 10 enables write of bits [27:24], and so on.	RW	0	-	-	X	X
3–0	RSS_IND_TABLE_ADDR	The indirection table is implemented as 16, 32-bit words. Each 32-bit word contains 8 receive queue IDs. Each receive queue ID is 4 bits wide.	RW	0	-	-	X	X

RLUP RSS DATA REGISTER (RLUP_RSS_DATA, OFFSET 0x204c)**Table 235: RLUP RSS Data Register (*rlup_rss_data*, offset 0x204c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X		
31–28	RSS_D7	In case of a RSS write this field contains RSS data that will be written into bits [31:28] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [31:28] of the indirection table row at the address specified in the RSS command register.	RW	0			X	X
27–24	RSS_D6	In case of a RSS write this field contains RSS data that will be written into bits [27:24] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [27:24] of the indirection table row at the address specified in the RSS command register.	RW	0			X	X

Table 235: RLUP RSS Data Register (rlup_rss_data, offset 0x204c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
23–20	RSS_D5	In case of a RSS write this field contains RSS data that will be written into bits [23:20] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [23:20] of the indirection table row at the address specified in the RSS command register.	RW	0	—	—	X	X
19–16	RSS_D4	In case of a RSS write this field contains RSS data that will be written into bits [19:16] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [19:16] of the indirection table row at the address specified in the RSS command register.	RW	0	—	—	X	X
15–12	RSS_D3	In case of a RSS write this field contains RSS data that will be written into bits [15:12] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [15:12] of the indirection table row at the address specified in the RSS command register.	RW	0	—	—	X	X
11–8	RSS_D2	In case of a RSS write this field contains RSS data that will be written into bits [11:8] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [11:8] of the indirection table row at the address specified in the RSS command register.	RW	0	—	—	X	X
7–4	RSS_D1	In case of a RSS write this field contains RSS data that will be written into bits [7:4] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [7:4] of the indirection table row at the address specified in the RSS command register.	RW	0	—	—	X	X
3–0	RSS_D0	In case of a RSS write this field contains RSS data that will be written into bits [3:0] of the indirection table at the address specified in the RSS command register. In case of a RSS read these field holds bits [3:0] of the indirection table row at the address specified in the RSS command register.	RW	0	—	—	—	X

RLUP FREE COUNT REGISTER (RLUP_FREE_COUNT, OFFSET 0x2074)*Table 236: RLUP Free Count Register (rlup_free_count, offset 0x2074)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–11	RESERVED		RO	0	X	X	X	X
10–0	RESERVED		RO	0	X	X	–	–
10–0	FREE_COUNT	This field indicates the number of unused CAM entries.	RO	0x400	–	–	X	X

RLUP IPv6 SOURCE ADDRESS 1 REGISTER (RLUP_IPV6_SRC1, OFFSET 0x2078)*Table 237: RLUP IPv6 Source Address 1 Register (rlup_ipv6_src1, offset 0x2078)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[127:96] of the IPv6 source address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register has no impact on a READ/WRITE operation.	RW	0	–	–	X	X

RLUP IPv6 SOURCE ADDRESS 2 REGISTER (RLUP_IPV6_SRC2, OFFSET 0x207c)*Table 238: RLUP IPv6 Source Address 2 Register (rlup_ipv6_src2, offset 0x207c)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[95:64] of the IPv6 source address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register has no impact on a READ/WRITE operation.	RW	0	–	–	X	X

RLUP IPv6 SOURCE ADDRESS 3 REGISTER (RLUP_IPV6_SRC3, OFFSET 0x2080)*Table 239: RLUP IPv6 Source Address 3 Register (rlup_ipv6_src3, offset 0x2080)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[63:32] of the IPv6 source address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register has no impact on a READ/WRITE operation.	RW	0	–	–	X	X

RLUP IPv6 SOURCE ADDRESS 4 REGISTER (RLUP_IPV6_SRC4, OFFSET 0x2084)*Table 240: RLUP IPv6 Source Address 4 Register (rlup_ipv6_src4, offset 0x2084)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[31:0] of the IPv6 source address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register has no impact on a READ/WRITE operation.	RW	0	–	–	X	X

RLUP IPv6 DESTINATION ADDRESS 1 REGISTER (RLUP_IPV6_DEST1, OFFSET 0x2088)*Table 241: RLUP IPv6 Destination Address 1 Register (rlup_ipv6_dest1, offset 0x2088)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[127:96] of the IPv6 destination address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register is has no impact on a READ/WRITE operation.	RW	0	-	-	X	X

RLUP IPv6 DESTINATION ADDRESS 2 REGISTER (RLUP_IPV6_DEST2, OFFSET 0x208C)*Table 242: RLUP IPv6 Destination Address 2 Register (rlup_ipv6_dest2, offset 0x208c)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[95:64] of the IPv6 destination address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register is has no impact on a READ/WRITE operation.	RW	0	-	-	X	X

RLUP IPv6 DESTINATION ADDRESS 3 REGISTER (RLUP_IPV6_DEST3, OFFSET 0x2090)*Table 243: RLUP IPv6 Destination Address 3 Register (rlup_ipv6_dest3, offset 0x2090)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[63:32] of the IPv6 destination address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register is has no impact on a READ/WRITE operation.	RW	0	-	-	X	X

RLUP IPv6 DESTINATION ADDRESS 4 REGISTER (RLUP_IPV6_DEST4, OFFSET 0x2094)*Table 244: RLUP IPv6 Destination Address 4 Register (rlup_ipv6_dest4, offset 0x2094)*

Description	Mode	Reset	06	08	09	16
These bits correspond to bits[31:0] of the IPv6 destination address. The write value of this register is used for ADD, INVALIDATE, and LOOKUP commands. This register is has no impact on a READ/WRITE operation.	RW	0	-	-	X	X

RLUP FTQ DATA REGISTERS (RLUP_FTQ_DATA[14], OFFSET 0x23C0)

Description	Mode	Reset	06	08	09	16
This register is used to access the FTQ data in the holding register within the FTQ for this block.	RW	0	X	X	X	X

RLUP FTQ COMMAND REGISTER (RLUP_FTQ_CMD, OFFSET 0x23F8)

The RX Lookup FTQ is 8 records deep.

Table 245: RLUP FTQ Command Register (rlup_ftq_cmd, Offset 0x23f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be read as 0 before the POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "RLUP FTQ Control Register (rlup_ftq_ctl, Offset 0x23fc)" on page 380. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "RLUP FTQ Control Register (rlup_ftq_ctl, Offset 0x23fc)" on page 380. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the current "RLUP FTQ Data Registers (rlup_ftq_data[14], Offset 0x23c0)" on page 618 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
(Unresolved for PG303) (Unresolved for PG203)								
27	ADD_INTERVENE	When this bit is written as a 1, the current "RLUP FTQ Data Registers (rlup_ftq_data[14], Offset 0x23c0)" on page 618 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
(Unresolved for PG303) (Unresolved for PG203)								

Table 245: RLUP FTQ Command Register (rlup_ftq_cmd, Offset 0x23f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16								
26	RD_DATA	When this bit is written as a 1, the "RLUP FTQ Data Registers (rlup_ftq_data[14], Offset 0x23c0)" on page 618 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X								
(Unresolved for PG303) (Unresolved for PG203)																
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X								
24-11	RESERVED		RO	0	X	X	X	X								
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the holding register (data).	RW	0	X	X	X	X								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>When written as 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When written as 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an intervene record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </tbody> </table>								Value	Name	Description	0	0	When written as 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When written as 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an intervene record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
Value	Name	Description														
0	0	When written as 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.														
1	1	When written as 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an intervene record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.														

Table 245: RLUP FTQ Command Register (rlup_ftq_cmd, Offset 0x23f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the "RLUP FTQ Control Register (rlup_ftq_ctl, Offset 0x23fc)" on page 380 .	RW	0	X	X	X	X

RLUP FTQ CONTROL REGISTER (RLUP_FTQ_CTL, OFFSET 0x23FC)*Table 246: RLUP FTQ Control Register (rlup_ftq_ctl, Offset 0x23fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the intervene bit on writes still works normally (See the ADD_INTERVENE of the “ RLUP FTQ Command Register (rlup_ftq_cmd, Offset 0x23f8) ” on page 376). When this bit is 0, the intervene input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the overflow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the intervene bit set and requires firmware intervention.	RO	0	X	X	X	X

RECEIVE VIRTUAL TO PHYSICAL COMMAND SCHEDULER (RV2PCS) REGISTERS

RV2PCS COMMAND REGISTER (RV2PCS_CH_COMMAND, OFFSET 0x2400)

Table 247: RV2PCS Command Register (rv2pcs_ch_command, offset 0x2400)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-1	RESERVED		RO	0	X	X	X	X
0	RESERVED		RO	0	X	X	-	-
0	ENABLE	This bit indicates the current enable status of this block. If this bit is set it indicates that the block is enabled. Clearing this bit has no effect.	RO	0	-	-	X	X

RV2PCS STATUS REGISTER (RV2PCS_CH_STATUS, OFFSET 0x2404)

Table 248: RV2PCS Status Register (rv2pcs_ch_status, offset 0x2404)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-18	RESERVED		RO	0	X	X	X	X
17-0	RESERVED		RO	0	X	X	-	-
17	SLOT_CNT_STAT	This bit is the current value of the generic CSCH_SLOT_CNT_STAT. This bit will toggle once each time a new slot is scheduled.	RO	0	-	-	X	X
16	CMD_CNT_STAT	This bit is the current value of the generic CSCH_CMD_CNT_STAT. This bit will toggle once each time FTQ entry is processed.	RO	0	-	-	X	X
15-0	RESERVED		RO	0	-	-	X	X

RECEIVE VIRTUAL TO PHYSICAL (RV2P) REGISTERS

The purpose of the Receive Virtual to Physical (RV2P) block is to determine the location in host memory where a block of data should be placed. It is implemented as a two-stage pipeline where the core of each pipeline is a programmable, register-based processor and associated logic. In the first stage, processor 1 accepts commands from its associated input FTQs and pre-fetches context fields, buffer descriptors, or page table entries and collects them into local cache for processor 2. The second processor is responsible for breaking up the host memory accesses into DMA operations which are performed by the Read DMA (RDMA) block.

RV2P COMMAND REGISTER (RV2P_COMMAND, OFFSET 0x2800)

Table 249: RV2P Command Register (rv2p_command, Offset 0x2800)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–19	RESERVED		RO	0	X	X	X	X
18	CTXIF_RESET	Setting this bit asserts the context interface soft-reset for one clock cycle.	WO	0	X	X	X	X
17	PROC2_RESET	Setting this bit asserts the processor 2 soft-reset for one clock cycle.	WO	0	X	X	X	X
16	PROC1_RESET	Setting this bit asserts the processor 1 soft-reset for one clock cycle.	WO	0	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X
9	ABORT5	This bit indicates that one of the two processors set its ABORT5 bit. When this bit is set, an attention will be generated and the processors may be stalled. The abort condition will remain until this bit is written as a one, at which time code execution will continue on both processors and the attention will be cleared.	WC	0	X	X	X	X
8	ABORT4	This bit indicates that one of the two processors set its ABORT4 bit. When this bit is set, an attention will be generated and the processors may be stalled. The abort condition will remain until this bit is written as a one, at which time code execution will continue on both processors and the attention will be cleared.	WC	0	X	X	X	X
7	ABORT3	This bit indicates that one of the two processors set its ABORT3 bit. When this bit is set, an attention will be generated and the processors may be stalled. The abort condition will remain until this bit is written as a one, at which time code execution will continue on both processors and the attention will be cleared.	WC	0	X	X	X	X

Table 249: RV2P Command Register (rv2p_command, Offset 0x2800) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
6	ABORT2	This bit indicates that one of the two processors set its ABORT2 bit. When this bit is set, an attention will be generated and the processors may be stalled. The abort condition will remain until this bit is written as a one, at which time code execution will continue on both processors and the attention will be cleared.	WC	0	X	X	X	X
5	ABORT1	This bit indicates that one of the two processors set its ABORT1 bit. When this bit is set, an attention will be generated and the processors may be stalled. The abort condition will remain until this bit is written as a one, at which time code execution will continue on both processors and the attention will be cleared.	WC	0	X	X	X	X
4	ABORT0	This bit indicates that one of the two processors set its ABORT0 bit. When this bit is set, an attention will be generated and the processors may be stalled. The abort condition will remain until this bit is written as a one, at which time code execution will continue on both processors and the attention will be cleared.	WC	0	X	X	X	X
3	RESERVED		RO	0	X	X	X	X
2	PROC2_INTRPT	Writing this bit asserts the processor 2 interrupt for one clock cycle. This bit will always read as 0.	WO	0	X	X	X	X
1	PROC1_INTRPT	Writing this bit asserts the processor 1 interrupt for one clock cycle. This bit will always read as 0.	WO	0	X	X	X	X
0	ENABLED	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the block is enabled (from the central Enable control registers). Writing this bit as a 0 has no effect. This bit is controlled by the central enable register.	RO	0	X	X	X	X

RV2P CONFIGURATION REGISTER (RV2P_CONFIG, OFFSET 0x2808)**Table 250: RV2P Configuration Register (rv2p_config, Offset 0x2808)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	X	X

Table 250: RV2P Configuration Register (rv2p_config, Offset 0x2808) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
27–24	PAGE_SIZE	This field sets the page size for each BD Page read by the RX BD Read and RX V2P Blocks.	RW	0x4	X	X	X	X
Value Name Description								
0	256	256-byte page, 16 BDs per page						
1	512	512-byte page, 32 BDs per page						
2	1K	1-KB page, 64 BDs per page						
3	2K	2-KB page, 128 BDs per page						
4	4K	4-KB page, 256 BDs per page						
5	8K	8-KB page, 512 BDs per page						
6	16K	16-KB page, 1024 BDs per page						
7	32K	32-KB page, 2048 BDs per page						
8	64K	64-KB page, 4096 BDs per page						
9	128K	128-KB page, 8192 BDs per page						
10	256K	256-KB page, 16384 BDs per page						
11	512K	512-KB page, 32768 BDs per page						
12	1m	1-MB page, 65536 BDs per page						
23–22	RESERVED		RO	0	X	X	X	X
21–16	PROC2_STALL_ON_ABORT	This field should be set to 0 for normal operation.	RW	0x3f	X	X	X	X
15–14	RESERVED		RO	0	X	X	X	X
13–8	PROC1_STALL_ON_ABORT	This field should be set to 0 for normal operation.	RW	0x3f	X	X	X	X
7–2	RESERVED		RO	0	X	X	X	X
1–0	STALL_PROC	This field should be set to 0 for normal operation.	RW	0x3	X	X	X	X

RV2P INSTRUCTION HIGH REGISTER (RV2P_INSTR_HIGH, OFFSET 0x2830)**Table 251: RV2P Instruction High Register (rv2p_instr_high, Offset 0x2830)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–5	RESERVED		RO	0	X	X	X	X
4–0	HIGH	This register contains the high-order instruction bits when writing or reading the processor instruction RAMs. High-order bits are bits [36–32].	RW	0	X	X	X	X

RV2P INSTRUCTION LOW REGISTER (RV2P_INSTR_LOW, OFFSET 0x2834)*Table 252: RV2P Instruction Low Register (rv2p_instr_low, Offset 0x2834)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	LOW	This register contains the low-order instruction bits when writing or reading the processor instruction RAMs. Low-order bits are bits [31–0].	RW	0	X	X	X	X

RV2P PROCESSOR 1 ADDRESS/COMMAND REGISTER (RV2P_PROC1_ADDR_CMD, OFFSET 0x2838)

Writing this register initiates a write or read of processor one's instruction memory. This should be performed only when the processor 1 stall bit is set in the config register.

Table 253: RV2P Processor 1 Address/Command Register (rv2p_proc1_addr_cmd, Offset 0x2838)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RDWR	This bit controls whether a write or read is performed of processor one's instruction RAM. A value of 1 initiates a write; a value of 0 initiates a read.	WO	0	X	X	X	X
30–10	RESERVED		RO	0	X	X	X	X
9–0	ADD	These bits contain the address in the processor one instruction RAM to write or read.	WO	0	X	X	X	X

RV2P PROCESSOR 2 ADDRESS/COMMAND REGISTER (RV2P_PROC2_ADDR_CMD, OFFSET 0x283c)

Writing this register initiates a write or read of processor two's instruction memory. This should be performed only when the processor 2 stall bit is set in the config register.

Table 254: RV2P Processor 2 Address/Command Register (rv2p_proc2_addr_cmd, Offset 0x283c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RDWR	This bit controls whether a write or read is performed of processor two's instruction RAM. A value of 1 initiates a write; a value of 0 initiates a read.	RW	0	X	X	X	X
30–10	RESERVED		RO	0	X	X	X	X
9–0	ADD	These bits contain the address in the processor two instruction RAM to write or read.	RW	0	X	X	X	X

RV2P PRIMARY FTQ DATA REGISTERS (RV2P_PFTQ_DATA[14], OFFSET 0x2B40)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	RO	0	X	X	X	X

RV2P Post FTQ COMMAND REGISTER (rv2p_pftq_cmd, Offset 0x2b78)

The depth of the RV2P Post FTQ is 16 records.

Table 255: RV2P Primary FTQ Command Register (rv2p_pftq_cmd, Offset 0x2b78)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the INTERVENE bit set as indicated in the “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the Intervene bit set as indicated in the “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the current “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVENE	When this bit is written as a 1, the current “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 255: RV2P Primary FTQ Command Register (rv2p_pftq_cmd, Offset 0x2b78) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
26	RD_DATA	When this bit is written as a 1, the “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389.	RW	0	X	X	X	X
Value Name Description								
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.						
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.						
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “RV2P Post FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)” on page 389.	RW	0	X	X	X	X

RV2P Post FTQ CONTROL REGISTER (RV2P_PFTQ_CTL, OFFSET 0x2B7C)*Table 256: RV2P Primary FTQ Control Register (rv2p_pftq_ctl, Offset 0x2b7c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE of the “ RV2P Post FTQ Command Register (rv2p_pftq_cmd, Offset 0x2b78) ” on page 387). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

RV2P TIMEOUT FTQ DATA REGISTERS (RV2P_TFTQ_DATA[14], OFFSET 0x2B80)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this block.	RO	0	X	X	X	X

RV2P TIMEOUT FTQ COMMAND REGISTER (RV2P_TFTQ_CMD, OFFSET 0x2BB8)

The depth of the RV2P Timeout FTQ is 32 records.

Table 257: RV2P Timeout FTQ Command Register (rv2p_tftq_cmd, Offset 0x2bb8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)" on page 392 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)" on page 392 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the current "RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)" on page 392 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the current "RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)" on page 392 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 257: RV2P Timeout FTQ Command Register (rv2p_ftfq_cmd, Offset 0x2bb8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
26	RD_DATA	When this bit is written as a 1, the “RV2P Timeout FTQ Control Register (rv2p_ftfq_ctl, Offset 0x2bbc)” on page 392 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “RV2P Timeout FTQ Control Register (rv2p_ftfq_ctl, Offset 0x2bbc)” on page 392.	RW	0	X	X	X	X
Value Name Description								
0 0 When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.								
1 1 When set to 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.								
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “RV2P Timeout FTQ Control Register (rv2p_ftfq_ctl, Offset 0x2bbc)” on page 392.	RW	0	X	X	X	X

RV2P TIMEOUT FTQ CONTROL REGISTER (RV2P_TFTQ_CTL, OFFSET 0x2BBC)*Table 258: RV2P Timeout FTQ Control Register (rv2p_tftq_ctl, Offset 0x2bbc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the intervene bit on writes still operates normally (See ADD_INTERVENE of the “RV2P Timeout FTQ Command Register (rv2p_tftq_cmd, Offset 0x2bb8)” on page 390). When this bit is 0, the INTERVENE input on the hardware interfaces operates normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the overflow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

RV2P MAILBOX FTQ DATA REGISTERS (RV2P_MFTQ_DATA[14], OFFSET 0x2BC0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this block.	RO	0	X	X	X	X

RV2P MAILBOX FTQ COMMAND REGISTER (rv2p_mftq_cmd, Offset 0x2bf8)

The depth of the RV2P Mailbox FTQ is 32 records.

Table 259: RV2P Mailbox FTQ Command Register (rv2p_mftq_cmd, Offset 0x2bf8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bf8)” on page 395. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the Intervene bit set as indicated in the “RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bf8)” on page 395. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the current “RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bf8)” on page 395 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVENE	When this bit is written as a 1, the current “RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bf8)” on page 395 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 259: RV2P Mailbox FTQ Command Register (rv2p_mftq_cmd, Offset 0x2bf8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16						
26	RD_DATA	When this bit is written as a 1, the “RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bfc)” on page 395 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X						
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X						
24–11	RESERVED		RO	0	X	X	X	X						
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl, Offset 0x2bfc)” on page 395.	RW	0	X	X	X	X						
Value Name Description														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">0</td> <td style="width: 10%;">0</td> <td>When set to 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When set to 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </table>									0	0	When set to 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When set to 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
0	0	When set to 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.												
1	1	When set to 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.												
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field.	RW	0	X	X	X	X						

RV2P MAILBOX FTQ CONTROL REGISTER (RV2P_MFTQ_CTL. OFFSET 0x2BFC)*Table 260: RV2P Mailbox FTQ Control Register (rv2p_mftq_ctl. Offset 0x2bfc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See the ADD_INTERVENE field of the "RV2P Mailbox FTQ Command Register (rv2p_mftq_cmd, Offset 0x2bf8)" on page 393). When this bit is 0, the INTERVENE input no the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

RECEIVE DMA (RDMA) REGISTERS

The purpose of the Receive DMA (RDMA) block is to move data received on the Ethernet interface and stored in on-chip memory to host memory as directed by RV2P. It also forwards data received on its input FTQ to the COM processor and performs CRC calculation/verification of iSCSI data.

RDMA COMMAND REGISTER (RDMA_COMMAND, OFFSET 0x2c00)

Table 261: RDMA Command Register (rdma_command, Offset 0x2c00)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–5	RESERVED		RO	0	X	X	X	X
4	MASTER_ABORT	This bit indicates that the current transaction received a master abort.	RO	0	X	X	—	—
4	RESERVED		RO	0	—	—	X	X
3–1	RESERVED		RO	0	X	X	X	X
0	ENABLED	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the block is enabled (from the “MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)” on page 232). Writing this bit as a 0 has no effect.	RO	0	X	X	X	X

RDMA STATUS REGISTER (RDMA_STATUS, OFFSET 0x2c04)

Table 262: RDMA Status Register (rdma_status, Offset 0x2c04)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–18	RESERVED		RO	0	X	X	—	—
31	CS16_ERR	When this bit is set it indicates that an error was detected in the CS16 value presented in the input FTQ.	WC	0	—	—	X	X
30–28	RESERVED		RO	0	—	—	X	X
27–20	ERR	This field is used for as an error indicator for internal debugging only and should never be set during normal operation.	RO	0	—	—	X	X
19–18	RESERVED		RO	0	—	—	X	X
17	BURST_CNT_STAT	This is the current status of the generic statistic output of the block. It toggles each time a new request is made to the central DMA block to cause a PCI transfer.	RO	0	X	X	X	X
16	DMA_CNT_STAT	This is the current status of the generic statistic output of the block. It toggles each time a new RX DMA FTQ entry is processed.	RO	0	X	X	X	X
15–3	RESERVED		RO	0	X	X	X	X



Table 262: RDMA Status Register (rdma_status, Offset 0x2c04) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2	CMP_FTQ_WAIT	This bit indicates that the current record in the FTQ is waiting because there is not enough room in the Completion FTQ to complete the DMA. This is not an error, but could indicate a problem downstream if set during chip idle state.	RO	0	X	X	X	X
1	MBUF_WAIT	This bit indicates that the current record in the FTQ is waiting because it can't get access to data from the RX MBUF.	RO	0	X	X	X	X
0	DMA_WAIT	This bit indicates that the current record in the FTQ is waiting because there are no available DMA channels. This is not an error, but could indicate a problem down-stream if set during chip idle state.	RO	0	X	X	X	X

RDMA CONFIGURATION REGISTER (RDMA_CONFIG, OFFSET 0x2c08)**Table 263: RDMA Configuration Register (rdma_config, Offset 0x2c08)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	–	–
31	DMA_BREAK_OVERRID E	When this bit is set the DMA_BREAKUP field is	RW	0	–	–	X	X
30	RESERVED		RO	0	–	–	X	X
29–28	DMA_BREAKUP	This field controls how RDMA should break up transactions to DMAE so that the byte enables are contiguous on the PCIe bus.	RW	0	–	–	X	X
Val Name Description								
0	00	Apply no alignment.						
1	01	Apply 4-byte alignment.						
2	10	Apply 8-byte alignment.						
3	11	Apply 8-byte alignment and then apply 4-byte alignment.						
27–18	CRC_OFFSET	This sets the 32-bit word offset into the TCP context where the RDMA CRC value can be found.	RW	0	–	–	X	X
17–11	RESERVED		RO	0	X	X	–	–
17	RESERVED		RO	0	–	–	X	X
16	CTXCACHE_DISABLE	Setting this bit disables the RDMA context cache.	RW	0	–	–	X	X
15–13	RESERVED		RO	0	–	–	X	X
12	DI_DISABLE	Setting this bit disables RDMA data integrity checksum checks.	RW	0	–	–	X	X
11	RESERVED		RO	0	–	–	X	X

Table 263: RDMA Configuration Register (rdma_config, Offset 0x2c08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
10–8	LINE_SZ	These bits control where the DMA alignment boundary (cache line size) is. This is the address granularity where DMA jobs larger or equal to the LIMIT_SZ size will be split.	RW	0x6	X	X	X	X
		Val ue Name Description						
		0 8 This setting is reserved and should not be used.						
		1 16 This setting is reserved and should not be used.						
		2 32 DMA jobs are aligned to 32-byte boundaries.						
		3 64 DMA jobs are aligned to 64-byte boundaries.						
		4 128 DMA jobs are aligned to 128-byte boundaries.						
		5 256 DMA jobs are aligned to 256-byte boundaries.						
		6 512 DMA jobs are aligned to 512-byte boundaries.						
7	RESERVED		RO	0	X	X	X	X
6–4	LIMIT_SZ	These bits control when the DMA alignment algorithm will be used. DMA jobs that are shorter than this value will not be aligned.	RW	0x6	X	X	–	–
		Val ue Name Description						
		0 8 This setting is reserved and should not be used.						
		1 16 This setting is reserved and should not be used.						
		2 32 DMA jobs less than 32 will not be aligned.						
		3 64 DMA jobs less than 64 will not be aligned.						
		4 128 DMA jobs less than 128 will not be aligned.						
		5 256 DMA jobs less than 256 will not be aligned.						
		6 512 DMA jobs less than 512 will not be aligned.						
6–4	RESERVED		RO	0	–	–	X	X
3	CACHE_ALIGN_EN	When this bit is set, the RX DMA block will align DMA operations to cache lines. Alignment can be controlled by the LIMIT_SZ and LINE_SZ fields.	RW	0	X	X	X	X

Table 263: RDMA Configuration Register (rdma_config, Offset 0x2c08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2	ONE_RECORD	When this bit is set, the RX DMA block will limit itself to processing one record at a time and will not look ahead into the FTQ to process multiple records at once. Multiple DMA requests may still be made on the same FTQ record. This is intended as a debug tool only.	RW	0	X	X	X	X
1-0	MAX_DMAS	These bits set the maximum number of DMA channels that can be allocated at any given time by RDMA. 1, 2, and 3 are valid values; a value of 0 will cause unpredictable RDMA behavior.	RW	0x2	X	X	-	-
1-0	RESERVED		RO	0	-	-	X	X

RDMA FTQ DATA REGISTERS (RDMA_FTQ_DATA[14], OFFSET 0x2FC0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.		0	X	X	X	X

RDMA FTQ COMMAND REGISTER (RDMA_FTQ_CMD, OFFSET 0x2FF8)

The depth of the RDMA FTQ is 16 records.

Table 264: RDMA FTQ Command Register (rdma_ftq_cmd, Offset 0x2ff8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the Intervene bit set as indicated in the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X

Table 264: RDMA FTQ Command Register (rdma_ftq_cmd, Offset 0x2ff8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
28	ADD_DATA	When this bit is written as a 1, the current “MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)” on page 232 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the current “MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)” on page 232 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)” on page 232 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges in-between the FTQ and the “MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)” on page 232.	RW	0	X	X	X	X
Value	Name	Description						
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.						
1	1	When set to 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.						

Table 264: RDMA FTQ Command Register (rdma_ftq_cmd, Offset 0x2ff8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field.	RW	0	X	X	X	X

RDMA FTQ CONTROL REGISTER (RDMA_FTQ_CTL, OFFSET 0x2FFC)**Table 265: RDMA FTQ Control Register (rdma_ftq_ctl, Offset 0x2ffc)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input no the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X

Table 265: RDMA FTQ Control Register (rdma_ftq_ctl, Offset 0x2ffc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

The purpose of the Receive Buffer Descriptor Cache (RBDC) block is to copy receive buffer descriptors from host memory to on-chip cache for RV2P to use.

RBDC CHECKSUM ERROR STATUS REGISTER (RBDC_CKSUM_ERROR_STATUS, OFFSET 0x3058)

Table 266: RBDC Checksum Error Status Register (rbdc_cksum_error_status, Offset 0x3058)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	RESERVED		RO	0	X	X	-	-
31-16	EXPECTED	This is the expected checksum for the data. This is the checksum that is sent by the block that provides the data.	RO	0	-	-	X	X
15-0	CALCULATED	This is the calculated checksum for the data. This is the checksum that is calculated locally over the data.	RO	0	-	-	X	X

RBDC DMA ERROR REGISTER (RBDC_DMA_ERROR, OFFSET 0x305C)

Table 267: RBDC DMA Error Register (rbdc_dma_error, Offset 0x305c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-0	RESERVED		RO	0	X	X	-	-
31	OCCURRED	When this bit is set it indicates that an error has occurred during a DMA transaction.	RO	0	-	-	X	X
30	RESERVED		RO	0	-	-	X	X
29-16	CID	This field shows the CID when the DMA error occurred.	RO	0	-	-	X	X
15-8	BDIDX	This field shows the BD index when the DMA error occurred.	RO	0	-	-	X	X
7-4	RESERVED		RO	0	-	-	X	X
3-0	LOWERIDX	This field shows the lower index of the cache when the DMA error occurred.	RO	0	-	-	X	X

MAILBOX QUEUE (MQ) REGISTERS

The mailbox queue (MQ) is responsible for receiving event indications from the host, storing the necessary information in context, and triggering the appropriate scheduler to handle the context update.

MQ COMMAND REGISTER (MQ_COMMAND, OFFSET 0x3c00)

Table 268: MQ Command Register (mq_command, Offset 0x3c00)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–7	RESERVED		RO	0	X	X	–	–
31–13	RESERVED		RO	0	–	–	X	X
12	NO_MAP_ERROR	This bit is set when a write was able to get through the write-mask, but was not able to find a mapping register value. This bit does not get forwarded to the attention output.	WC	0	–	–	X	X
11	NO_BIN_ERROR	This bit is set when a write is made to a CID that cannot be converted to a bin number. This bit does not get forwarded to the attention output.	WC	0	–	–	X	X
10	IDB_OVERFLOW	This bit is set when the IDB overflow state is entered. This bit is forwarded to the attention output. Clearing this bit can be accomplished to remove the MQ from the IDB overflow state. This bit may also be cleared automatically based on a FIFO threshold when the idb_drop_auto_recov bit is set.	WC	0	–	–	X	X
9–8	RESERVED		RO	0	–	–	X	X
7	IDB_CFG_ERROR	When this bit is set, it indicates that there is a misconfiguration in the IDB buffer size, location in the mailbox, or location in context. All address manipulation for the feature is accomplished using address muxing/shifting, so these sizes and locations must accommodate. This value will clear when the values are programmed properly.	RO	0	–	–	X	X
6	RD_ERROR	When this bit is set, it indicates that a read was made to the mailbox queue area on the PCI Bus “MQ Bad Read Address Register (mq_bad_rd_addr, Offset 0x3c18)” on page 408 . Writing this bit to a 1 will clear the RD_ERROR status. Writing this bit as a 0 has no effect. When this bit is set, the attention output of the mailbox queue block will be asserted.	WC	0	X	X	–	–

Table 268: MQ Command Register (mq_command, Offset 0x3c00) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
6	RD_ERROR	When this bit is set, it indicates that a read was made to the mailbox queue area on the PCI Bus. While this bit is set and the HALT_DIS field of the "MQ Configuration Register (mq_config, Offset 0x3c08)" on page 405 is not set, all reads will be accepted and dropped. No reads will be made from context. The "MQ Bad Read Address Register (mq_bad_rd_addr, Offset 0x3c18)" on page 408 will indicate the address of the access. Writing this bit to a 1 will clear the RD_ERROR status. Writing this bit as a 0 has no effect. When this bit is set, the attention output of the mailbox queue block will be asserted.	WC	0	X	X	X	X
5	WR_ERROR	When this bit is set, it indicates that an illegal write was made to the mailbox queue area on the PCI Bus. While this bit is set and the HALT_DIS field of the "MQ Configuration Register (mq_config, Offset 0x3c08)" on page 405 is not set, all writes will be accepted and dropped. No writes will be made to context "MQ Bad Write Address Register (mq_bad_wr_addr, Offset 0x3c14)" on page 408. Writing this bit to a 1 will clear the WR_ERROR status. Writing this bit as a 0 has no effect. When this bit is set, the attention output of the mailbox queue block will be asserted.	WC	0	X	X	X	X
4	OVERFLOW	When this bit is set, it indicates that a request was made to write the MBQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up at some time since the bit was last cleared. Writing this bit to a 1 will clear the overflow status. This is not an error indication, but rather a debug tool. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
3–2	RESERVED		RO	0	X	X	X	X
1	RESERVED		RO	0	X	X	—	—
1	INIT	When this bit is set it will cause the index storage and scanner storage to be erased. This is automatically done at reset.	SC	0x1	—	—	X	X

Table 268: MQ Command Register (mq_command, Offset 0x3c00) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	ENABLED	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the block is enabled (from the central Enable control registers). Writing this bit as a 0 has no effect. This bit is controlled by the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232.	RO	0	X	X	X	X

MQ STATUS REGISTER (MQ_STATUS, OFFSET 0x3c04)**Table 269: MQ Status Register (mq_status, Offset 0x3c04)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–20	RESERVED		RO	0				
19	RESERVED		RO	0	X	X	—	—
19	IDB_OFLOW_STAT	This bit is a generic statistic that toggles each time the DB Overflow state is entered.	RO	0	—	—	X	X
18	PCI_STALL_STAT	This is a generic statistic output that toggles each time 64 PCI clocks have occurred where the mailbox queue is back-pressuring the PCI bus. This will happen when the FIFO is full.	RO	0	X	X	X	X
17	CTX_ACCESS64_STAT	This is a generic statistic output that toggles each time the context is written with a single 64-bit word.	RO	0	X	X	X	X
16	CTX_ACCESS_STAT	This is a generic statistic output that toggles each time the context is written with a single cycle word. The value may be 64 or 32 bits.	RO	0	X	X	X	X
15–0	RESERVED		RO	0	X	X	X	X

MQ CONFIGURATION REGISTER (MQ_CONFIG, OFFSET 0x3c08)**Table 270: MQ Configuration Register (mq_config, Offset 0x3c08)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–27	RESERVED		RO	0	X	X	X	X
26–20	CUR_DEPTH	This value indicates the current Mailbox Queue depth being used. A value of 0 indicates that the Mailbox Queue is empty.	RO	0	X	X	X	X
19–15	RESERVED		RO	0	X	X	X	X

Table 270: MQ Configuration Register (mq_config, Offset 0x3c08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16												
14-8	MAX_DEPTH	This value controls the amount of the Mailbox Queue that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. The only reason to shorten the queue is to reserve some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up.	RW	0x40	X	X	X	X												
7	RESERVED		RO	0	X	X	X	X												
6-4	KNL_BYP_BLK_SIZE	This value controls the size of the kernel bypass windows. Normally, this is set to 4 KB to match the page size of the system, but can be set smaller for systems with limited PCI address space.	RW	0x4	X	X	X	X												
		<table border="1"> <thead> <tr> <th>Value Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0 256</td><td>256-Byte Kernel Bypass Pages.</td></tr> <tr> <td>1 512</td><td>512-Byte Kernel Bypass Pages.</td></tr> <tr> <td>2 1k</td><td>1024-Byte Kernel Bypass Pages.</td></tr> <tr> <td>3 2k</td><td>2048-Byte Kernel Bypass Pages.</td></tr> <tr> <td>4 4k</td><td>4096-Byte Kernel Bypass Pages.</td></tr> </tbody> </table>	Value Name	Description	0 256	256-Byte Kernel Bypass Pages.	1 512	512-Byte Kernel Bypass Pages.	2 1k	1024-Byte Kernel Bypass Pages.	3 2k	2048-Byte Kernel Bypass Pages.	4 4k	4096-Byte Kernel Bypass Pages.						
Value Name	Description																			
0 256	256-Byte Kernel Bypass Pages.																			
1 512	512-Byte Kernel Bypass Pages.																			
2 1k	1024-Byte Kernel Bypass Pages.																			
3 2k	2048-Byte Kernel Bypass Pages.																			
4 4k	4096-Byte Kernel Bypass Pages.																			
3-2	RESERVED		RO	0	X	X	-	-												
3	DIS_IDB_DROP	When this bit is set it causes the MQ to not drop IDB data writes. Setting this bit may lead to deadlocks with the system.	RW	0	-	-	X	X												
2	BIN_MQ_MODE	When this bit is set the MQ will bypass the CID to Bin mapping function in the "enlist" statem machine. This will allow the bin number to directly determine what connection each MA mailbox area is associated with. This reduces the total required size fo the mailbox BAR space from 128MB to 16MB. When this mode is enabled, the bin number will be used for IDB tracking.	RW	0	-	-	X	X												

Table 270: MQ Configuration Register (mq_config, Offset 0x3c08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	HALT_DIS	When this bit is clear, the Mailbox queue will go into a drop state when the WR_ERROR bit is set and stays until the WR_ERROR bit is cleared. While in the drop state the mailbox queue drops all writes made to it from the PCI bus to protect the context contents from corruption. When this bit is set, the drop state is disabled and the Mailbox queue will continue to process PCI writes normally when even when the WR_ERROR bit is set. Note: This bit should be set for BCM5709 AX controllers to prevent OS initiated reads of the "Mailbox Queue Data Register (pcicfg_mailbox_queue_data, Offset 0x94)" on page 178 from setting the RD_ERROR bit of the "MQ Command Register (mq_command, Offset 0x3c00)" on page 403.	RW	0	X	X	X	X
0	TX_HIGH_PRI	When this bit is set, TX Activate messages will be enqueued as high priority; otherwise, they will be enqueued as normal priority.	RW	0	X	X	X	X

MQ ENQUEUE 1 REGISTER (MQ_ENQUEUE1, OFFSET 0x3c0c)

Table 271: MQ Enqueue 1 Register (mq_enqueue1, Offset 0x3c0c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–29	RESERVED		RO	0	X	X	X	X
28	KNL_MODE	This field selects what mask registers will be consulted for the next GRC enqueue operation. If this bit is 0, then the kernel bypass masks will be used. If this bit is 1, then the kernel masks will be used.	RW	0	X	X	X	X
27–24	BYTE_MASK	This field controls which bytes will be written to the mailbox. A bit must be 1 to enable writing of the corresponding byte in the "MQ Enqueue 2 Register (mq_enqueue2, Offset 0x3c10)" on page 408.	RW	0	X	X	X	X
23–22	RESERVED		RO	0	X	X	X	X
21–8	CID	This value sets the context that will be updated when the "MQ Enqueue 2 Register (mq_enqueue2, Offset 0x3c10)" on page 408 is written with 32 bits of data. This CID value is based on the 256-byte spacing, similar to the mailbox PCI address space.	RW	0	X	X	X	X

Table 271: MQ Enqueue 1 Register (mq_enqueue1, Offset 0x3c0c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7–2	OFFSET	This value sets the Offset within the context that will be updated when the “MQ Enqueue 2 Register (mq_enqueue2, Offset 0x3c10)” on page 408 is written with 32 bits of data.	RW	0	X	X	X	X
1–0	RESERVED		RO	0	X	X	X	X

MQ ENQUEUE 2 REGISTER (MQ_ENQUEUE2, OFFSET 0x3c10)

Description	Mode	Reset	06	08	09	16
This register will enqueue 32 bits of data (along with the CID and OFFSET specified in RW “MQ Enqueue 1 Register (mq_enqueue1, Offset 0x3c0c)” on page 407) into the Mailbox Queue when written. When the entry is pulled from the queue, it will be operated on normally with all the special indications for the Offset specified. No checking is performed on this write, so if no mask values are set for the Offset specified, unpredictable actions may result. If the queue is full, then this register write must block until the write has completed. When this register is read, it will return the last value written.		0	X	X	X	X

MQ BAD WRITE ADDRESS REGISTER (MQ_BAD_WR_ADDR, OFFSET 0x3c14)

Description	Mode	Reset	06	08	09	16
This register is loaded when the WR_ERROR field of “MQ Command Register RO (mq_command, Offset 0x3c00)” on page 403 is set in the command register. This may be used to attempt to detect which software piece incorrectly wrote the NetXtreme II. If a subsequent write occurs before WR_ERROR is cleared, it will be ignored. This register will hold the address of the first bad write.	0	X	X	X	X	X

MQ BAD READ ADDRESS REGISTER (MQ_BAD_RD_ADDR, OFFSET 0x3c18)

Description	Mode	Reset	06	08	09	16
This register is loaded when the RD_ERROR field of the “MQ Command Register RO (mq_command, Offset 0x3c00)” on page 403 is set. This may be used to attempt to detect what piece of software read the NetXtreme II incorrectly. If a subsequent read occurs before RD_ERROR is cleared, it will be ignored. This register will hold the address of the first bad read.	0	X	X	X	X	X

MQ KERNEL BYPASS WINDOW START REGISTER (MQ_KNL_BYP_WIND_START, OFFSET 0x3c1c)

Table 272: MQ Kernel Bypass Window Start Register (mq_knl_byp_wind_start, Offset 0x3c1c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–12	VALUE	This value indicates the PCI address (Offset from the BAR value) where the Kernel Bypass Mailbox Window pages start. All accesses below this value will be interpreted as 256-byte kernel Mailbox Windows. The size of the kernel bypass window is controlled by the KNL_BYP_BLK_SIZE field of the “MQ Configuration Register (mq_config, Offset 0x3c08)” on page 405.	RW	0x40				
11–0	RESERVED		RO	0				

MQ KERNEL WINDOWS END REGISTER (MQ_KNL_WIND_END, OFFSET 0x3c20)

Table 273: MQ Kernel Windows End Register (mq_knl_wind_end, Offset 0x3c20)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–8	VALUE	This value provides an accurate end check for the kernel (256-byte) Mailbox windows. The value is an Offset from the PCI BAR value. All valid kernel windows must be less than this value. If VALUE is less than the “MQ Kernel Bypass Window Start Register (mq_knl_byp_wind_start, Offset 0x3c1c)” on page 409, then the WR_ERROR or RD_ERROR bits will be set for any access between the two values. If VALUE is greater than the “MQ Kernel Bypass Window Start Register (mq_knl_byp_wind_start, Offset 0x3c1c)” on page 409, the kernel bypass area will be exposed in the area between the pointers.	RW	0x400	X	X	X	X
7–0	RESERVED		RO	0	X	X	X	X

MQ KERNEL WRITE MASK 1 REGISTER (MQ_KNL_WRITE_MASK1, OFFSET 0x3c24)

Description	Mode	Reset	06	08	09	16
This register provides a mask that allows each word within the first 128-byte Mailbox window to be enabled for normal write. When a bit is set, the corresponding word is enabled for write. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. When a write is made to a kernel mailbox window and the corresponding mask bit is not set, it is considered a write error and the WR_ERROR bit of the “MQ Command Register (mq_command, Offset 0x3c00)” on page 403 is set.	RW	0xffffffff	X	X	X	X



MQ KERNEL TRANSMIT MASK 1 REGISTER (MQ_KNL_TX_MASK1, OFFSET 0x3c28)

Description	Mode	Reset	06	08	09	16
This register will cause the TX Scheduler to be activated when any word enabled by RW this mask is written. For the word to be written, the corresponding bit in the "MQ Kernel Write Mask 1 Register (mq_knl_write_mask1, Offset 0x3c24)" on page 409 must also be set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the first 128 bytes of the Mailbox window.	0	X	X	-	-	-

MQ KERNEL COMMAND MASK 1 REGISTER (MQ_KNL_CMD_MASK1, OFFSET 0x3c2c)

Description	Mode	Reset	06	08	09	16
This register will cause the Command Scheduler to be activated when any word RW enabled by this mask is written. For the word to be written, the corresponding bit in the "MQ Kernel Write Mask 1 Register (mq_knl_write_mask1, Offset 0x3c24)" on page 409 must also be set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the first 128 bytes of the Mailbox window.	0	X	X	-	-	-

MQ KERNEL CONDITIONAL ENQUEUE MASK 1 REGISTER (MQ_KNL_COND_ENQUEUE_MASK1, OFFSET 0x3c30)

Description	Mode	Reset	06	08	09	16
This register will cause a conditional enqueue operation. Only the even bits in this word RW may be set. Setting the odd bits may cause unexpected behavior. When a conditional enqueue is requested, byte 15–8 must be enabled for write. First, the context value for that byte corresponding byte in the next word will be read. Normally, this will be tcp_prod_retx_num. Second, the read value will be compared to the written value written. Third, if the values differ, no further action will be taken. If the values are the same, then the original byte write will occur. Normally, this is to position tcp_cons_retx_num. Also the scheduler will be enqueued with the CID for the connection being modified. This will start the transmitter back up after it has been reloaded during a retransmit. In summary, the driver writes its consumer retx number to the tcp_cons_retx_num and the mailbox compares this write to the tcp_prod_retx_num and only executes the write (and enqueue) if they match. The corresponding bit in the "MQ Kernel Write Mask 1 Register (mq_knl_write_mask1, Offset 0x3c24)" on page 409 bit must never be set when this bit is set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the first 128 bytes of the Mailbox window.	0	X	X	-	-	

MQ KERNEL RECEIVE V2P MASK 1 REGISTER (MQ_KNL_RX_V2P_MASK1, OFFSET 0x3c34)

Description	Mode	Reset	06	08	09	16
This register will cause the RX V2P to be activated when any word enabled by this RW mask is written. For the word to be written, the knl_write_mask must also be set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the first 128 bytes of the Mailbox window.	0	X X - -				

MQ KERNEL WRITE MASK 2 REGISTER (MQ_KNL_WRITE_MASK2, OFFSET 0x3c38)

Description	Mode	Reset	06	08	09	16
This register provides a mask that allows each word within the first 128-byte Mailbox RW window to be enabled for normal write. When a bit is set, the corresponding word is enabled for write. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. When a write is made to a kernel mailbox window and the corresponding mask bit is not set, it is considered a write error and the WR_ERROR bit of the "MQ Command Register (mq_command, Offset 0x3c00)" on page 403 is set.	0	X X X X				

MQ KERNEL TRANSMIT MASK 2 REGISTER (MQ_KNL_TX_MASK2, OFFSET 0x3c3c)

Description	Mode	Reset	06	08	09	16
This register will cause the TX Scheduler to be activated when any word enabled by this mask is written. For the word to be written, the corresponding bit in the "MQ Kernel Write Mask 2 Register (mq_knl_write_mask2, Offset 0x3c38)" on page 411 must also be set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the second 128 bytes of the Mailbox window.	0x10	X X - -				

MQ KERNEL COMMAND MASK 2 REGISTER (MQ_KNL_CMD_MASK2, OFFSET 0x3c40)

Description	Mode	Reset	06	08	09	16
This register will cause the Command Scheduler to be activated when any word enabled by this mask is written. For the word to be written, the corresponding bit in the "MQ Kernel Write Mask 2 Register (mq_knl_write_mask2, Offset 0x3c38)" on page 411 must also be set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the second 128 bytes of the Mailbox window.	0	X X - -				

MQ KERNEL CONDITIONAL ENQUEUE MASK 2 REGISTER (MQ_KNL_COND_ENQUEUE_MASK2, OFFSET 0x3c44)

Description	Mode	Reset	06	08	09	16
This register will cause a conditional enqueue operation. Only the even bits in this word RW may be set. Setting the odd bits may cause unexpected behavior. When a conditional enqueue is requested, byte 15–8 must be enabled for write. First, the context value for that byte corresponding byte in the next word will be read. Normally, this will be tcp_prod_retx_num. Second, the read value will be compared to the written value written. Third, if the values differ, no further action will be taken. If the values are the same, then the original byte write will occur. Normally, this is to position tcp_cons_retx_num. Also the scheduler will be enqueued with the CID for the connection being modified. This will start the transmitter back up after it has been reloaded during a retransmit. In summary, the driver writes its consumer retx number to the tcp_cons_retx_num and the mailbox compares this write to the tcp_prod_retx_num and only executes the write (and enqueue) if they match. The corresponding bit in the “MQ Kernel Write Mask 2 Register (mq_knl_write_mask2, Offset 0x3c38)” on page 411 bit must never be set when this bit is set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the first 128 bytes of the Mailbox window.	0	X	X	–	–	

MQ KERNEL RECEIVE V2P MASK 2 REGISTER (MQ_KNL_RX_V2P_MASK2, OFFSET 0x3c48)

Description	Mode	Reset	06	08	09	16
This register will cause the RV2P to be activated when any word enabled by this mask RW is written. For the word to be written, the knl_write_mask must also be set. This value is used when the kernel area is written. A separate value is used when the kernel bypass area is written. This value handles the second 128 bytes of the Mailbox window.	0	X	X	–	–	–

MQ KERNEL BYPASS WRITE MASK 1 REGISTER (MQ_KNL_BYP_WRITE_MASK1, OFFSET 0x3c4c)

Description	Mode	Reset	06	08	09	16
This register operates the same as the “MQ Kernel Write Mask 1 Register (mq_knl_write_mask1, Offset 0x3c24)” on page 409 , except this value is used for the first 128 bytes of a kernel bypass area. When a write is made to a kernel bypass mailbox window and the corresponding mask bit is not set, it is considered a write error and the WR_ERROR bit of the “MQ Command Register (mq_command, Offset 0x3c00)” on page 403 is set.	0	X	X	X	X	X

MQ KERNEL BYPASS TRANSMIT MASK 1 REGISTER (MQ_KNL_BYP_TX_MASK1, OFFSET 0x3c50)

Description	Mode	Reset	06	08	09	16
This register operates the same as the “MQ Kernel Transmit Mask 1 Register RW (mq_knl_tx_mask1, Offset 0x3c28)” on page 410, except this value is used for the first 128 bytes of a kernel bypass area.	0	X X - -				

MQ KERNEL BYPASS COMMAND MASK 1 REGISTER (MQ_KNL_BYP_CMD_MASK1, OFFSET 0x3c54)

Description	Mode	Reset	06	08	09	16
This register operates the same as the “MQ Kernel Command Mask 1 Register RW (mq_knl_cmd_mask1, Offset 0x3c2c)” on page 410, except this value is used for the first 128 bytes of a kernel bypass area.	0	X X - -				

MQ KERNEL BYPASS CONDITIONAL ENQUEUE MASK 1 REGISTER (MQ_KNL_BYP_COND_ENQUEUE_MASK1, OFFSET 0x3c58)

Description	Mode	Reset	06	08	09	16
This register operates the same as the knl_cond_byte_clr_mask, except this value is RW used for the first 128 bytes of a kernel bypass area.	0	X X - -				

MQ KERNEL BYPASS RECEIVE V2P MASK 1 REGISTER (MQ_KNL_BYP_RX_V2P_MASK1, OFFSET 0x3c5c)

Description	Mode	Reset	06	08	09	16
This register operates the same as the “MQ Kernel Receive V2P Mask 1 Register RW (mq_knl_rx_v2p_mask1, Offset 0x3c34)” on page 411, except this value is used for the first 128 bytes of a kernel bypass area.	0	X X - -				

MQ KERNEL BYPASS WRITE MASK 2 REGISTER (MQ_KNL_BYP_WRITE_MASK2, OFFSET 0x3c60)

Description	Mode	Reset	06	08	09	16
This register operates the same as the “MQ Kernel Write Mask 1 Register RW (mq_knl_write_mask1, Offset 0x3c24)” on page 409, except this value is used for the second 128 bytes of a kernel bypass area. When a write is made to a kernel bypass mailbox window and the corresponding mask bit is not set, it is considered a write error and the WR_ERROR bit of the “MQ Command Register (mq_command, Offset 0x3c00)” on page 403 is set.	0	X X X X				

MQ KERNEL BYPASS TRANSMIT MASK 2 REGISTER (MQ_KNL_BYP_TX_MASK2, OFFSET 0x3c64)

Description	Mode	Reset	06	08	09	16
This register operates the same as the "MQ Kernel Transmit Mask 1 Register RW (mq_knl_tx_mask1, Offset 0x3c28)" on page 410, except this value is used for the second 128 bytes of a kernel bypass area.	0	X	X	-	-	-

MQ KERNEL BYPASS COMMAND MASK 2 REGISTER (MQ_KNL_BYP_CMD_MASK2, OFFSET 0x3c68)

Description	Mode	Reset	06	08	09	16
This register operates the same as the MQ Kernel Command Mask 1 Register RW (mq_knl_cmd_mask1, Offset 0x3c2c), except this value is used for the second 128 bytes of a kernel bypass area.	0	X	X	-	-	-

MQ KERNEL BYPASS CONDITIONAL ENQUEUE MASK 2 REGISTER (MQ_KNL_BYP_COND_ENQUEUE_MASK2, OFFSET 0x3c6c)

Description	Mode	Reset	06	08	09	16
This register operates the same as the knl_cond_byte_clr_mask, except this value is RW used for the second 128 bytes of a kernel bypass area.	0	X	X	-	-	-

MQ KERNEL BYPASS RECEIVE V2P MASK 2 REGISTER (MQ_KNL_BYP_RX_V2P_MASK2, OFFSET 0x3c70)

Description	Mode	Reset	06	08	09	16
This register operates the same as the MQ Kernel Receive V2P Mask 1 Register RW (mq_knl_rx_v2p_mask1, Offset 0x3c34) on page 411, except this value is used for the second 128 bytes of a kernel bypass area.	0	X	X	-	-	-

MQ MEMORY WRITE ADDRESS REGISTER (MQ_MEM_WR_ADDR, OFFSET 0x3c74)

Table 274: MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–6	RESERVED		RO	0	X	X	X	X
5–0	VALUE	This address value is used for the GRC write diagnostic access the internal FIFO ram as described in the mem_wr_data registers below.	RW	0	X	X	X	X

MQ MEMORY WRITE DATA 0 REGISTER (MQ_MEM_WR_DATA0, OFFSET 0x3c78)*Table 275: MQ Memory Write Data 0 Register (mq_mem_wr_data0, Offset 0x3c78)*

Description	Mode	Reset	06	08	09	16
This register is used to write data pointed to by the VALUE field of the “MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74)” on page 414. Writes to this register go to a holding register that will be written to bits 31 down to 0 when the “MQ Memory Write Data 2 Register (mq_mem_wr_data2, Offset 0x3c80)” on page 415 is written.	WO	0	X	X	X	X

MQ MEMORY WRITE DATA 1 REGISTER (MQ_MEM_WR_DATA1, OFFSET 0x3c7c)*Table 276: MQ Memory Write Data 1 Register (mq_mem_wr_data1, Offset 0x3c7c)*

Description	Mode	Reset	06	08	09	16
This register is used to write data pointed to by the VALUE field of the “MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74)” on page 414. Writes to this register go to a holding register that will be written to bits 63 down to 32 when the “MQ Memory Write Data 2 Register (mq_mem_wr_data2, Offset 0x3c80)” on page 415 is written.	WO	0	X	X	X	X

MQ MEMORY WRITE DATA 2 REGISTER (MQ_MEM_WR_DATA2, OFFSET 0x3c80)*Table 277: MQ Memory Write Data 2 Register (mq_mem_wr_data2, Offset 0x3c80)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30	RESERVED		RO	0	X	X	—	—
30–0	VALUE	This register is used to write data pointed to by the VALUE field of the “MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74)” on page 414. Writes cause the data written to all mem_wr_data registers to be written to the mailbox ram. The data in this register is written.			—	—	X	X
29–0	VALUE	This register is used to write data pointed to by the VALUE field of the “MQ Memory Write Address Register (mq_mem_wr_addr, Offset 0x3c74)” on page 414. Writes cause the data written to all mem_wr_data registers to be written to the mailbox ram. The data in this register is written.	RW	0	X	X	—	—

MQ MEMORY READ ADDRESS REGISTER (MQ_MEM_RD_ADDR, OFFSET 0x3C84)*Table 278: MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–6	RESERVED		RO	0	X	X	X	X
5–0	VALUE	This address value is used for the GRC diagnostic read access the internal FIFO ram as described in the mem_rd_data registers below.	RW	0	X	X	X	X

MQ MEMORY READ DATA 0 REGISTER (MQ_MEM_RD_DATA0, OFFSET 0x3C88)*Table 279: MQ Memory Read Data 0 Register (mq_mem_rd_data0, Offset 0x3c88)*

Description	Mode	Reset	06	08	09	16
This register is used to read data bits 31–0 pointed to by the VALUE field of the “MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84)” on page 416. Reads of this register initiate an actual read of the RAM data.	RO	0	X	X	X	X

MQ MEMORY READ DATA 1 REGISTER (MQ_MEM_RD_DATA1, OFFSET 0x3C8C)*Table 280: MQ Memory Read Data 1 Register (mq_mem_rd_data1, Offset 0x3c8c)*

Description	Mode	Reset	06	08	09	16
This register is used to read data bits 63–32 pointed to by the VALUE field of the “MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84)” on page 416. The “MQ Memory Read Data 0 Register (mq_mem_rd_data0, Offset 0x3c88)” on page 416 must be read before this one to make this value valid.	RO	0	X	X	X	X

MQ MEMORY READ DATA 2 REGISTER (MQ_MEM_RD_DATA2, OFFSET 0x3c90)*Table 281: MQ Memory Read Data 2 Register (mq_mem_rd_data2, Offset 0x3c90)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	RESERVED		RO	0	X	X	X	X
30	RESERVED		RO	0	X	X	-	-
30–0	VALUE	This register is used to read data bits 93–64 pointed to by the VALUE field of the “MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84)” on page 416. The “MQ Memory Read Data 0 Register (mq_mem_rd_data0, Offset 0x3c88)” on page 416 must be read before this one to make this value valid.			-	-	X	X
29–0	VALUE	This register is used to read data bits 93–64 pointed to by the VALUE field of the “MQ Memory Read Address Register (mq_mem_rd_addr, Offset 0x3c84)” on page 416. The “MQ Memory Read Data 0 Register (mq_mem_rd_data0, Offset 0x3c88)” on page 416 must be read before this one to make this value valid.	RO	0	X	X	-	-

MQ DEBUG VECTOR PEEK REGISTER (MQ_DEBUG_VECT_PEEK, OFFSET 0x3c94)*Table 282: MQ Debug Vector Peek Register (mq_debug_vect_peek, offset 0x3c94)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	-	-
31–28	2_SEL	When 2_en is '1', this value controls what debug vector will be displayed in 2_value.	RW	0	-	-	X	X
27	2_EN	When this bit is '1', the debug vector control is taken over from the misc block and controlled by the 2_sel value.	RW	0	-	-	X	X
26–16	2_VALUE	Value of selected debug vector for debug bus #2.	RO	0	-	-	X	X
15–12	1_SEL	When 1_en is '1', this value controls what debug vector will be displayed in 1_value.	RW	0	-	-	X	X
11	1_EN	When this bit is '1', the debug vector control is taken over from the misc block and controlled by the 1_sel value.	RW	0	-	-	X	X
10–0	1_VALUE	Value of selected debug vector for debug bus #1.	RO	0	-	-	X	X



MQ IMMEDIATE DATA BUFFER CONFIGURATION REGISTER (MQ_IDB_CFG, OFFSET 0x3CA0)*Table 283: MQ Immediate Data Buffer Configuration Register (mq_idb_cfg, offset 0x3ca0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	This bit must be set to '1' to enable the immediate data buffer feature of the MQ block.	RW	0	–	–	X	X
30	Reserved		RO	0	–	–	X	X
29–24	TRIG_LOC	This value configures the offset in the kernel bypass mailbox window that will be treated as the trigger that moves an IDB into "full" state. The value specifies the 32-bit word within the first 256 bytes of mailbox window that will be treated this way. The driver should write the SQIDX of the WQE for the immediate data into this location as a 32-bit write. The MQ will automatically replace the upper 8 bits of this write with the immediate data buffer number that was allocated before the 32-bit value is written to context. Mapping feature may be used to map this value to locations above the first 256B of context. The mask feature may be used to enable this address to generate a FTQ entry for a particular processor. This position must also be enabled for write in the knl_byp_write_mask1/2 registers.	RW	0xa	–	–	X	X
23–21	Reserved		RO	0	–	–	X	X
20–10	CTX_LOC	This value sets the location in context where the immediate data buffers are located. This memory must be allocated from the context as MB will write to this area when the IDB window areas are of the kernel bypass windows are written. This value must be aligned to the size of the buffer chosen in mb_size and the area allocated must be large enough to account for the 4 supported immediate data buffers. If mb_size is set to 256B, then this value may be set to any value to locate the buffers in context virtual memory space. 1 KB of context memory must be allocated. If mb_size is set to 512B, then bit 0 of this value must be zero. 2 KB of context memory must be allocated. If mb_size is set to 1024B, then bit 0 and 1 of this value must be zero. 4 KB of context memory must be allocated. If mb_size is set to 2048B, then bit 0, 1, and 2 of this value must be zero. 8 KB of context memory must be allocated.	RW	0	–	–	X	X

Table 283: MQ Immediate Data Buffer Configuration Register (*mq_idb_cfg*, offset 0x3ca0) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
9-8	WQE_SIZE	This value controls the size of the WQE area within the IDB window. The wQE area may be disabled so that the entire IDB window is data area by setting this value to '0'. This setting only controls the area over which the add_swap will not applied within the IDB window.	RW	0x1	-	-	X	X
Value Name Description								
	0 NONE	No WQE are at start of IDB window. Entire IDB window is subject to additional swapping.						
	1 64B	64B WQE size. The first 64 bytes of the IDB window will not be swapped additionally.						
	2 128B	128B WQE size. The first 128 bytes of the IDB window will not be swapped additionally.						
7	ADD_WORD_SWAP	This value controls additional word swap done within the data portion of the IDB window. This swapping is done in addition to swapping configured for the mailbox queue in the pci block. This setting allows the swap of the data portion of a WQE+data message to be swapped differently than the WQE portion. The WQE portion will be swapped like normal mailbox write data.	RW	0	-	-	X	X
6	ADD_BYTE_SWAP	This value controls additional byte swap done within the data portion of the IDB window. This swapping is done in addition to swapping configured for the mailbox queue in the pci block. This setting allows the swap of the data portion of a WQE+data message to be swapped differently than the WQE portion. The WQE portion will be swapped like normal mailbox write data.	RW	0x1	-	-	X	X

Table 283: MQ Immediate Data Buffer Configuration Register (*mq_idb_cfg*, offset 0x3ca0) (Cont.)

Bit	Name	Description			Mode	Reset	06	08	09	16
5–4	MB_SIZE	Enumeration:			RW	0	–	–	X	X
Value Name Description										
		0	256	This value sets the size of the IDB window and the size of the buffer in context memory to 256B.						
		1	512	This value sets the size of the IDB window and the size of the buffer in context memory to 512B.						
		2	1K	This value sets the size of the IDB window and the size of the buffer in context memory to 1024B.						
		3	2K	This value sets the size of the IDB window and the size of the buffer in context memory to 2048B.						
3–2	Reserved				RO	0	–	–	X	X
1–0	MB_START	Enumeration:			RW	0	–	–	X	X
Value Name Description										
		0	256	The mailbox for IDB data writes starts at 256B into the kernel bypass window. The IDB size must be set to 256 if this size is chosen.						
		1	512	The mailbox for IDB data writes starts at 512B into the kernel bypass window. The IDB size must be set to 512B or less if this size is chosen.						
		2	1K	The mailbox for IDB data writes starts at 1024B into the kernel bypass window. The IDB size must be set to 1 KB or less if this size is chosen.						
		3	2K	The mailbox for IDB data writes starts at 2048B into the kernel bypass window.						

MQ IDB FREE REGISTER (MQ_IDB_FREE, OFFSET 0x3CA4)*Table 284: MQ IDB Free Register (mq_idb_free, offset 0x3ca4)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	X	X	X	X
20–7	Reserved		RO	0	X	X	—	—
20–7	CID	When this value is written, any IDB not in the "idle" state that is assigned to this CID will be set to "idle" state. This will allow the IDB to be used again by this or a different CID. Note: The IDB state can not move directly from some of the states to the "idle" state. This is honored when this register is written, but the IDB will eventually be returned to the "free" state where it may be reused.	RW	0	—	—	X	X
6–0	Reserved		RO	0	X	X	X	X

MQ IDB STATE 0 VALUE REGISTER (MQ_IDB_STATE0_VAL, OFFSET 0x3CB0)*Table 285: MQ IDB State 0 Value Register (mq_idb_state0_val, offset 0x3cb0)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	X	X	X	X
20–7	RESERVED		RO	0	X	X	—	—
20–7	CID	This value indicates the CID that has allocated this IDB. This value is only valid if the state is not "idle".	RO	0	—	—	X	X
6–3	RESERVED		RO	0	X	X	X	X
2–0	RESERVED		RO	0	X	X	—	—
2–0	STATE	This field is the current state of this IDB. This value can be written to "idle" to release the IDB manually. The IDB state can not move directly from some of the states to the "idle" state. This is honored when this register is written to the "idle" state. The state will return to "idle" when the trigger write has left the internal MQ FIFO: if this state written to zero ("idle"), the read value may move to the "outwait" state before returning to the "idle" state.	RW	0	—	—	X	X

Table 285: MQ IDB State 0 Value Register (*mq_idb_state0_val*, offset 0x3cb0) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2-0 (cont.)	STATE	The only valid write value for this field is "idle" (zero). All other write values will be ignored.	RW	0	-	-	X	X
		Value	Name	Description				
		0	IDLE	This IDB is "idle" which means it may be allocated if a write is made to the IDB starting mailbox address.				
		1	FILLING	This IDB is "filling" which means it has been allocated, but has not been triggered by writing to the IDB trigger mailbox address.				
		2	TRIGGERED	This IDB is "triggered" which means that the trigger location write has entered the MQ FIFO. When in this state, the FSM must pass through "outwait" to return to the "idle" state without going through the "full" state.				
		3	FULL	This IDB is "full" which means it has been allocated and filled by the host driver, and the trigger write has reached the context, and the IDB is ready to be used by CP and TXP.				
		4	OUTWAIT	This IDB is "outwait" which means that the IDB is in the process of being recovered (returned to "idle"), but a trigger write still exists in the FIFO. When the trigger write leaves the FIFO, the IDB will return to "idle" from this state.				

MQ CONFIGURATION 2 REGISTER (MQ_CONFIG2, OFFSET 0x3D00)*Table 286: MQ Configuration 2 Register (mq_config2, offset 0x3d00)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–21	RESERVED		RO	0	X	X	X	X
20	RESERVED		RO	0	X	X	—	—
20	SCNR_CTHRU_DIS	When this bit is set to '1', the cut-through mode for the scanner sub-block is disabled and all bin numbers will be routed through the dirty bits array. When this bit is zero, the scanner block will bypass the array if the array is currently empty and the output of the scanner is empty.	RW	0	—	—	X	X
19	RESERVED		RO	0	X	X	X	X
18–16	RESERVED		RO	0	X	X	—	—
18–17	IDB_AUTO_ON	Enumeration:	RW	0	—	—	X	X
		Value Name Description						
		0 32 Exit IDB drop mode when FIFO has less than 32 entries in it.						
		1 16 Exit IDB drop mode when FIFO has less than 16 entries in it.						
		2 8 Exit IDB drop mode when FIFO has less than 8 entries in it.						
		3 4 Exit IDB drop mode when FIFO has less than 4 entries in it.						
16	IDB_DROP_AUTO_REC OV	When set to '1', the idb overflow attention will automatically clear when the FIFO falls below the idb_auto_on threshold.	RW	0	—	—	X	X
15–13	RESERVED		RO	0	X	X	X	X
12–8	RESERVED		RO	0	X	X	—	—
12–8	FIRST_L4L5	This is the first CID that will be used for L4/L5 context. This CID will map to bin# 256. This value has a granularity of 256 CIDs (value of 2 is CID 512). A value of zero is an invalid setting as this will conflict with the L2 CID locations.	RW	0x1	—	—	X	X
7	RESERVED		RO	0	X	X	X	X
6–4	RESERVED		RO	0	X	X	—	—

Table 286: MQ Configuration 2 Register (*mq_config2*, offset 0x3d00) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
6–4	CONT_SZ	Enumeration:	RW	0x5	X	X	–	–
		Value Name Description						
	2	4PER 4 CIDs to one container.						
	3	6PER 6 CIDs to one container.						
	4	8PER 8 CIDs to one container.						
	5	10PER 10 CIDs to one container.						
	6	12PER 12 CIDs to one container.						
	7	14PER 14 CIDs to one container.						
3–0	Reserved		RO	0	X	X	X	X

MQ INDEX COMMAND REGISTER (MQ_IDX_CMD, OFFSET 0x3D04)**Table 287: MQ Index Command Register (*mq_idx_cmd*, offset 0x3d04)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RRESERVED		RO	0	X	X	X	X
27–16	RESERVED		RO	0	X	X	–	–
27–16	BIN	This value controls what bit is accessed in the index storage RAM. This value must be written with the rd_cmd/wr_cmd value.	RW	0	–	–	X	X
15–14	Reserved		RO	0	X	X	X	X
13–12	RESERVED		RO	0	X	X	–	–
13–12	BIN_OFFSET	This value sets bits [3:2] of the bin_offset for each access. This value must be written with the rd_cmd/wr_cmd value.	RW	0	–	–	X	X
11–6	RESERVED		RO	0	X	X	X	X
5–4	RESERVED		RO	0	X	X	–	–
5–4	SP	These bits are used to set and read the special bits associated with the 32-bit value in the idx_data register. On writes, these bits must be set when the wr_cmd is written non-zero. On reads, these bits will be valid when the rd_cmd bit returns to zero.	RW	0	–	–	X	X
3	RESERVED		RO	0	X	X	X	X
2–0	RESERVED		RO	0	X	X	–	–

Table 287: MQ Index Command Register (*mq_idx_cmd*, offset 0x3d04) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2-1	WR_CMD	When this bit is written as a 2'b01, 2'b10, or 2'b11, the current idx_data is written to the bin and bin-offset specified in this register. Both these bits and the rd_cmd bits must be '0' when this field is written as non-zero.	SC	0	-	-	X	X
Value								
0	NOTHIN G	No request.						
1	LOW	Write data[15:0] and sp[0].						
2	HIGH	Write data[31:16] and sp[1].						
3	BOTH	Write data[31:0] and sp[1:0].						
0	RD_CMD	When this bit is written as a '1', value at the bin and bin-offset specified in this register will be read into the idx_data register. Both this bit and the wr_cmd bits must be '0' when this bit is written as a '1'. The read data will be valid after this bit returns to '0'.	SC	0	-	-	X	X

MQ INDEX DATA REGISTER (MQ_IDX_DATA, OFFSET 0x3D08)

Description	Mode	Reset	06	08	09	16
Data to be written to index storage must be written here before the "MQ Index Command Register (<i>mq_idx_cmd</i> , offset 0x3d04)" on page 424. Data read using the "MQ Index Command Register (<i>mq_idx_cmd</i> , offset 0x3d04)" on page 424 will be visible here after the RD_CMD field is read as 0.	RW	0	-	-	X	X

MQ SCANNER COMMAND REGISTER (MQ_SCNR_CMD, OFFSET 0x3D0C)Table 288: MQ Scanner Command Register (*mq_scnr_cmd*, offset 0x3d0c)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-28	RESERVED		RO	0	X	X	X	X
27-16	RESERVED		RO	0	X	X	-	-
27-16	BIN	This value controls what bin is accessed in the scanner storage. This value must be written with the rd_cmd/wr_cmd value.	RW	0	-	-	X	X
15-2	RESERVED		RO	0	X	X	X	X
1-0	RESERVED		RO	0	X	X	-	-

Table 288: MQ Scanner Command Register (mq_scnr_cmd, offset 0x3d0c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	WR_CMD	When this bit is written as a '1', the current scnr_data[0] bit is written to the bin specified in this register. This bit and the rd_cmd bit must be '0' when this field is written as a '1'.	SC	0	-	-	X	X
0	RD_CMD	When this bit is written as a '1', values for the 32 bins at bin[11:5] specified in this register will be read into the scnr_data register. Both this bit and the wr_cmd bit must be '0' when this bit is written as a '1'. The read data will be valid after this bit returns to '0'.	SC	0	-	-	X	X

MQ SCANNER DATA REGISTER (MQ_SCNR_DATA, OFFSET 0x3D10)

Description	Mode	Reset	06	08	09	16
Data to be written to scanner storage must be written to bit zero of this register before RW the "MQ Scanner Command Register (mq_scnr_cmd, offset 0x3d0c)" on page 425.	0	-	-	X	X	
Data read using the "MQ Scanner Command Register (mq_scnr_cmd, offset 0x3d0c)" on page 425 will be visible here after the RD_CMD field is read as 0.						

MQ MAP L2_0 REGISTER (MQ_MAP_L2_0, OFFSET 0x3D20)**Table 289: MQ Map L2_0 Register (mq_map_l2_0, offset 0x3d20)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	-	-
31	ENA	When this bit is set the rule is enabled.	RW	0x1	-	-	X	X
30–28	RESERVED		RO	0	-	-	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x2	-	-	X	X
Value	Name	Description						
0	NONE	No FTQ write.						
1	TSCH	Write to TSCHQ.						
2	CS	Write to CSQ.						
3	RV2PCS	Write to RV2PCSQ.						

Table 289: MQ Map L2 0 Register (*mq_map_l2_0*, offset 0x3d20) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16																				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0	–	–	X	X																				
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>8-bit or 16-bit index</td></tr> <tr> <td>1</td><td>1</td><td>8-bit or 16-bit index</td></tr> <tr> <td>2</td><td>2</td><td>32-bit index</td></tr> <tr> <td>4</td><td>4</td><td>8-bit or 16-bit index</td></tr> <tr> <td>5</td><td>5</td><td>8-bit or 16-bit index</td></tr> <tr> <td>6</td><td>6</td><td>8-bit, 16-bit, or 32-bit index</td></tr> </tbody> </table>	Value	Name	Description	0	0	8-bit or 16-bit index	1	1	8-bit or 16-bit index	2	2	32-bit index	4	4	8-bit or 16-bit index	5	5	8-bit or 16-bit index	6	6	8-bit, 16-bit, or 32-bit index					
Value	Name	Description																										
0	0	8-bit or 16-bit index																										
1	1	8-bit or 16-bit index																										
2	2	32-bit index																										
4	4	8-bit or 16-bit index																										
5	5	8-bit or 16-bit index																										
6	6	8-bit, 16-bit, or 32-bit index																										
22–20	RESERVED		RO	0	–	–	X	X																				
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x1	–	–	X	X																				
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X																				
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>1</td><td>8B</td><td>8-bit value</td></tr> <tr> <td>2</td><td>16B</td><td>16-bit value</td></tr> <tr> <td>3</td><td>32B</td><td>32-bit value</td></tr> </tbody> </table>	Value	Name	Description	1	8B	8-bit value	2	16B	16-bit value	3	32B	32-bit value														
Value	Name	Description																										
1	8B	8-bit value																										
2	16B	16-bit value																										
3	32B	32-bit value																										
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x6	–	–	X	X																				

MQ MAP L2 1REGISTER (MQ_MAP_L2_1, OFFSET 0x3d24)**Table 290: MQ Map L2 1 Register (mq_map_l2_1, offset 0x3d24)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x1	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x90	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x8a	–	–	X	X

MQ MAP L2 2 REGISTER (MQ_MAP_L2_2, OFFSET 0x3d28)*Table 291: MQ Map L2 2 Register (mq_map_l2_2, offset 0x3d28)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x1	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x92	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x3	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x90	–	–	X	X

MQ MAP L2 3 REGISTER (MQ_MAP_L2_3, OFFSET 0x3D2C)*Table 292: MQ Map L2 3 Register (mq_map_l2_3, offset 0x3d2c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x4	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x1	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x6	–	–	X	X

MQ MAP L2 4 REGISTER (MQ_MAP_L2_4, OFFSET 0x3d30)*Table 293: MQ Map L2 4 Register (mq_map_l2_4, offset 0x3d30)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x5	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0	–	–	X	X

MQ MAP L2 5 REGISTER (MQ_MAP_L2_5, OFFSET 0x3d34)**Table 294: MQ Map L2 5 Register (mq_map_l2_5, offset 0x3d34)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x6	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x2	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x3	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x8	–	–	X	X

MQ MAP L4 0 REGISTER (MQ_MAP_L4_0, OFFSET 0x3d40)*Table 295: MQ Map L4 0 Register (mq_map_l4_0, offset 0x3d40)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x72	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x1	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0xb8	–	–	X	X

MQ MAP L4 1 REGISTER (MQ_MAP_L4_1, OFFSET 0x3d44)*Table 296: MQ Map L4 1 Register (mq_map_l4_1, offset 0x3d44)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x1	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x90	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x8a	–	–	X	X



MQ MAP L4 2 REGISTER (MQ_MAP_L4_2, OFFSET 0x3d48)*Table 297: MQ Map L4 2 Register (mq_map_l4_2, offset 0x3d48)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x1	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x92	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x3	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x90	–	–	X	X



MQ MAP L4_3 REGISTER (MQ_MAP_L4_3, OFFSET 0x3D4C)*Table 298: MQ Map L4_3 Register (mq_map_l4_3, offset 0x3d4c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x4	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0	–	–	X	X

MQ MAP L4 4 REGISTER (MQ_MAP_L4_4, OFFSET 0x3d50)*Table 299: MQ Map L4 4 Register (mq_map_l4_4, offset 0x3d50)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x5	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x70	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0xb2	–	–	X	X

MQ MAP L4_5 REGISTER (MQ_MAP_L4_5, OFFSET 0x3d54)**Table 300: MQ Map L4_5 Register (mq_map_l4_5, offset 0x3d54)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x3	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x6	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0xb4	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x3	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0xb4	–	–	X	X



MQ MAP L5_0 REGISTER (MQ_MAP_L5_0, OFFSET 0x3D60)*Table 301: MQ Map L5_0 Register (mq_map_l5_0, offset 0x3d60)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0	–	–	X	X

MQ MAP L5_1 REGISTER (MQ_MAP_L5_1, OFFSET 0x3d64)*Table 302: MQ Map L5_1 Register (mq_map_l5_1, offset 0x3d64)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x1	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x21	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x6	–	–	X	X



MQ MAP L5 2 REGISTER (MQ_MAP_L5_2, OFFSET 0x3D68)*Table 303: MQ Map L5 2 Register (mq_map_l5_2, offset 0x3d68)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x20	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x3	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0	–	–	X	X



MQ MAP L5_3 REGISTER (MQ_MAP_L5_3, OFFSET 0x3D6C)*Table 304: MQ Map L5_3 Register (mq_map_l5_3, offset 0x3d6c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x4	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0	–	–	X	X



MQ MAP L5 4 REGISTER (MQ_MAP_L5_4, OFFSET 0x3D70)*Table 305: MQ Map L5 4 Register (mq_map_l5_4, offset 0x3d70)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies the FTQ that will be written to when this index is updated in context.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within the 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x5	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0x22	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x2	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0xa	–	–	X	X

MQ MAP L5_5 REGISTER (MQ_MAP_L5_5, OFFSET 0x3d74)*Table 306: MQ Map L5_5 Register (mq_map_l5_5, offset 0x3d74)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	–	–
31	ENA	When this bit is set the rule is enabled.	RW	0x1	–	–	X	X
30–28	RESERVED		RO	0	–	–	X	X
27–26	ARM	Specifies FTQ that will be written to when this index is updated in context. For this mapping register, the CP is activated.	RO	0x2	–	–	X	X
		Value	Name	Description				
		0	NONE	No FTQ write.				
		1	TSCH	Write to TSCHQ.				
		2	CS	Write to CSQ.				
		3	RV2PCS	Write to RV2PCSQ.				
25–23	BIN_OFFSET	Designates the offset within 16-byte bin where the index will be stored. This is read-only, and can not be adjusted. This means that there is one mapping register for each index location, for each connection type.	RO	0x6	–	–	X	X
		Value	Name	Description				
		0	0	8-bit or 16-bit index				
		1	1	8-bit or 16-bit index				
		2	2	32-bit index				
		4	4	8-bit or 16-bit index				
		5	5	8-bit or 16-bit index				
		6	6	8-bit, 16-bit, or 32-bit index				
22–20	RESERVED		RO	0	–	–	X	X
19–10	CTX_OFFSET	Offset that will be used for the context write of the index. This value is used for bits [11:2] of the context address. The bits below that for the context write are defined by the MQ_OFFSET[1:0] values.	RO	0xa	–	–	X	X
9–8	SZ	Size of the index stored in the index bin.	RO	0x3	–	–	X	X
		Value	Name	Description				
		1	8B	8-bit value				
		2	16B	16-bit value				
		3	32B	32-bit value				
7–0	MQ_OFFSET	Specifies the address in the mailbox that the host writes to for the index (after normal mailbox swapping configured in the TGT block). This is a byte address, but the address must be aligned to the SZ of the index. MQ_OFFSET[0] must be 0 for 16-bit indexes. MQ_OFFSET[1:0] must be 2'b00 for 32-bit indexes.	RO	0x28	–	–	X	X

The purpose of the Command Scheduler (CSCH) is to schedule commands to the Command Processor (CP). The CSCH keeps track of all connections that need to be processed and schedules them one-by-one to the CP.



CSCH FTQ DATA REGISTERS (csch_ch_ftq_data[14], Offset 0x43c0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ RW for this block.		0	X	X	X	X

CSCH FTQ COMMAND REGISTER (csch_ch_ftq_cmd, Offset 0x43f8)

The Command Scheduler FTQ is 32 records deep.

Table 307: CSCH FTQ Command Register (csch_ch_ftq_cmd, Offset 0x43f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the INTERVENE bit set as indicated in the “CSCH FTQ Control Register (csch_ch_ftq_ctl, Offset 0x43fc)” on page 447. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “CSCH FTQ Control Register (csch_ch_ftq_ctl, Offset 0x43fc)” on page 447. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the current “CSCH FTQ Data Registers (csch_ch_ftq_data[14], Offset 0x43c0)” on page 707 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the current “CSCH FTQ Data Registers (csch_ch_ftq_data[14], Offset 0x43c0)” on page 707 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 307: CSCH FTQ Command Register (csch_ch_ftq_cmd, Offset 0x43f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
26	RD_DATA	When this bit is written as a 1, the "CSCH FTQ Data Registers (csch_ch_ftq_data[14], Offset 0x43c0)" on page 707 is updated with the data from the FTQ entry pointed to by the OFFSET field. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the "CSCH FTQ Data Registers (csch_ch_ftq_data[14], Offset 0x43c0)" on page 707.	RW	0	X	X	X	X
Value Name Description								
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.						
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET field. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.						
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field.	RW	0	X	X	X	X

CSCH FTQ CONTROL REGISTER (CSCH_CH_FTQ_CTL, OFFSET 0x43FC)*Table 308: CSCH FTQ Control Register (csch_ch_ftq_ctl, Offset 0x43fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the INTERVENE bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

The purpose of the Timer is to keep track of active connections, checking the corresponding TCP timer values for timeouts and enqueueing events to the Completion Processor (COM) as well as providing a set of general-purpose counters/timers that can be used by software/firmware.

TSCH FTQ DATA REGISTERS (TSCH_FTQ_DATA[14], OFFSET 0x4FE0)

Description	Mode	Reset	06	08	09	16
This register is used to access the FTQ data in the holding register within the FTQ for RO	0	X X X X				

TSCH FTQ COMMAND REGISTER (TSCH_FTQ_CMD, OFFSET 0x4FF8)

The TX Scheduler FTQ is 32 records deep.

Table 309: TSCH FTQ Command Register (tsch_ftq_cmd, Offset 0x4ff8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4fc)" on page 449 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4fc)" on page 449 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the "TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4fc)" on page 449 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the "TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4fc)" on page 449 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the "TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4fc)" on page 449 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X



Table 309: TSCH FTQ Command Register (tsch_ftq_cmd, Offset 0x4ff8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16						
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X						
24–11	RESERVED		RO	0	X	X	X	X						
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “ TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4ffc) ” on page 449.	RW	0	X	X	X	X						
Value Name Description														
<table border="0"> <tr> <td style="vertical-align: top; padding-right: 10px;">0</td> <td style="vertical-align: top; padding-right: 10px;">0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">1</td> <td style="vertical-align: top; padding-right: 10px;">1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </table>									0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.												
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.												
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field.	RW	0	X	X	X	X						

TSCH FTQ CONTROL REGISTER (TSCH_FTQ_CTL, OFFSET 0x4FFC)**Table 310: TSCH FTQ Control Register (tsch_ftq_ctl, Offset 0x4ffc)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X

Table 310: TSCH FTQ Control Register (*tsch_ftq_ctl*, Offset 0x4ffc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue: <ul style="list-style-type: none">• It can decrease total chip latency.• It reserves some space in the queue so that old entries can be viewed after a hang, because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the INTERVENE bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the INTERVENE status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

TRANSMIT BUFFER DESCRIPTOR READ (TBDR) REGISTERS

The Buffer Descriptor Read (TBDR) block is responsible for fetching buffer descriptor (BD's) or page table entries (PTE's) from host necessary to fill transmit slots that are generated by the Transmit Scheduler (TSCH). In addition, TBDR also write the BD's or PTE's to the Transmit Buffer Descriptor Cache (TBDC).

TBDR COMMAND REGISTER (TBDR_COMMAND, OFFSET 0x5000)

Table 311: TBDR Command Register (tbdr_command, Offset 0x5000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–5	RESERVED		RO	0	X	X	X	X
4	MSTR_ABORT	When this bit reads as a 1, the TBDR block has received a master abort from the DMA engine. The status for the abort is in the status register below. By writing this bit as a 1, the TX BD Cache block will be forced to complete an aborted DMA operation. This will result in bogus BD's in the cache. This should be done only as a work-around attempt. If this bit is set, TBDR_ATTN will be asserted. While in this state, the Host Coalescing block will not free the DMA channel where the error was encountered.	WC	0	X	X	X	X
3–2	RESERVED		RO	0	X	X	X	X
1	SOFT_RST	When this bit is written as a 1, the TBDR will execute a soft reset. This should never be done during operation but is intended for use as a work around for testing. This bit will always read as a 0.	SC	0	X	X	X	X
0	ENABLE	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the block is enabled. Writing this bit as a 0 has no effect. This bit is controlled by the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232 .	RO	0	X	X	X	X

TBDR STATUS REGISTER (TBDR_STATUS, OFFSET 0x5004)*Table 312: TBDR Status Register (tbdr_status, Offset 0x5004)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31-7	RESERVED		RO	0	X	X	X	X
6	BURST_CNT	This bit is a generic statistic for the number of DMA transactions have been processed by the TBDR. This bit toggles once each time a request is made to the DMA block to cause a PCI transfer.	RO	0	X	X	X	X
5	FTQ_ENTRY_CNT	This bit is a generic statistic for the number of FTQ entries that have been processed by the TBDR. This bit toggles once each time a FTQ entry is processed.	RO	0	X	X	X	X
4	SEARCHMISS_ERROR	This bit indicates that tbdr_camwrite got a miss on a search that was supposedly already written to the cache. This should never happen. The bit can be cleared only by a hard or soft reset of the tbdr block.	RO	0	X	X	X	X
3	FIFO_UNDERFLOW	This bit indicates that the FIFO between tbdr_camwrite and tbdr_bdpipe underflowed. Once this bit is set, it will remain set until a hard or soft reset occurs. Operation of the FIFO will be unpredictable if this bit is set. The bit should never be set in normal operation.	RO	0	X	X	X	X
2	FIFO_OVERFLOW	This bit indicates that the FIFO between tbdr_camwrite and tbdr_bdpipe overflowed. Once this bit is set, it will remain set until a hard or soft reset occurs. Operation of the FIFO will be unpredictable if this bit is set. The bit should never be set in normal operation.	RO	0	X	X	X	X
1	FTQ_WAIT	This bit indicates that TBDR is waiting for the TX Processor FTQ to write before moving on when it reads as 1. This bit toggles during normal operation, so it is only valid during idle or lock-up conditions.	RO	0	X	X	X	X
0	DMA_WAIT	This bit indicates that TBDR is waiting for the DMA to complete before moving on when it reads as 1. This bit toggles during normal operation, so it is only valid during idle or lock-up conditions.	RO	0	X	X	X	X

TBDR CONFIGURATION REGISTER (TBDR_CONFIG, OFFSET 0x5008)*Table 313: TBDR Configuration Register (tbdr_config, Offset 0x5008)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	X	X
27–24	PAGE_SIZE	This field sets the page size for each BD Page read by the TBDR block. This register also configures other TX BD readers such as the TX DMA.	RW	0x4	X	X	X	X
		Val Name Description						
		0 256 256-byte page, 16 BDs per page						
		1 512 512-byte page, 32 BDs per page						
		2 1K 1-KB page, 64 BDs per page						
		3 2K 2-KB page, 128 BDs per page						
		4 4K 4-KB page, 256 BDs per page						
		5 8K 8-KB page, 512 BDs per page						
		6 16K 16-KB page, 1024 BDs per page						
		7 32K 32-KB page, 2048 BDs per page						
		8 64K 64-KB page, 4096 BDs per page						
		9 128K 128-KB page, 8192 BDs per page						
		10 256K 256-KB page, 16384 BDs per page						
		11 512K 512-KB page, 32768 BDs per page						
		12 1m 1-MB page, 65536 BDs per page						
23–11	RESERVED		RO	0	X	X	X	X
10	CACHE_NEXT_PA GE_PTRS	If this bit is set, then TX BD Read will write next page pointers to the TX BD Cache.	RW	0	X	X	X	X
9	PRIORITY	This bit sets the PCI priority of DMA operations performed by the cache. What does a 1 mean?	RW	0	X	X	X	X
8	SWAP_MODE	This bit sets the DMA swap mode for read operations performed by TX BD Read. What does a 1 mean?	RW	0	X	X	X	X
7–0	MAX_BDS	This field limits the number of BD's that will be read for a slot. The goal of this register is to prevent a bad application from consuming the entire TX BD Cache and stopping all other traffic. Setting this value to 0 is invalid.	RW	0x40	X	X	X	X

TBDR FTQ DATA REGISTERS (TBDR_FTQ_DATA[14], OFFSET 0x53c0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ RO for this block.		0	X	X	X	X

TBDR FTQ COMMAND REGISTER (TBDR_FTQ_CMD, OFFSET 0x53F8)

The depth of the TX BD Read FTQ is 4 records.

Table 314: TBDR FTQ Command Register (tbdr_ftq_cmd, Offset 0x53f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “ TBDR FTQ Control Register (tbdr_ftq_ctl, Offset 0x53fc) ” on page 738. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the Intervene bit set as indicated in the “ TBDR FTQ Control Register (tbdr_ftq_ctl, Offset 0x53fc) ” on page 738. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the “ TBDR FTQ Data Registers (tbdr_ftq_data[14], Offset 0x53c0) ” on page 735 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the “ TBDR FTQ Data Registers (tbdr_ftq_data[14], Offset 0x53c0) ” on page 735 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “ TBDR FTQ Data Registers (tbdr_ftq_data[14], Offset 0x53c0) ” on page 735 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 314: TBDR FTQ Command Register (tbdr_ftq_cmd, Offset 0x53f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “ TBDR FTQ Data Registers (tbdr_ftq_data[14], Offset 0x53c0) ” on page 735.	RW	0	X	X	X	X
Value Name								
		Description						
		0 0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.					
		1 1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.					
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field.	RW	0	X	X	X	X

TBDR FTQ CONTROL REGISTER (TBDR_FTQ_CTL, OFFSET 0x53FC)*Table 315: TBDR FTQ Control Register (tbdr_ftq_ctl, Offset 0x53fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input no the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

TRANSMIT DMA (TDMA) REGISTERS

The DMA (TDMA) block is responsible for moving data from host memory to the Payload Queue, calculating a checksum over the data, and inserting holes in the payload if present on per-packet basis. The DMA is located at the end of the transmit pipeline. This transmit pipeline relies on the shadowing of the BD information to achieve parallelism and the DMA is also responsible to update the shadow BD information in other two transmit stages, i.e., Scheduler and BD Read.

TDMA COMMAND REGISTER (TDMA_COMMAND, OFFSET 0x5c00)

Table 316: TDMA Command Register (tdma_command, Offset 0x5c00)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–20	RESERVED		RO	0	X	X	–	–
31	IFIFO_CLR	When this bit is set the overflow condition of the input FIFO is cleared.	SC	0	–	–	X	X
30	OFIFO_CLR	When this bit is set the overflow condition of the output FIFO is cleared.	SC	0	–	–	X	X
29–25	RESERVED		RO	0	–	–	X	X
24	FORCE_ILOCK_CKERR	When this bit is set an error is forced into the interlocking checksum calculation.	RW	0	–	–	X	X
23	MASK_CS4	When this bit is set the fourth component of TDMA checksum error is masked.	RW	0	–	–	X	X
22	MASK_CS3	When this bit is set the third component of TDMA checksum error is masked.	RW	0	–	–	X	X
21	MASK_CS2	When this bit is set the second component of TDMA checksum error is masked.	RW	0	–	–	X	X
20	MASK_CS1	When this bit is set the first component of TDMA checksum error is masked.	RW	0	–	–	X	X
19–8	RESERVED		RO	0	–	–	X	X
7	BAD_L2_LENGTH_ABORT	This bit indicates that the current transaction specifies a L2 packet that did not end on a BD application boundary. This is always a fatal error and should never happen. When this bit is set, the TX DMA block will wait for it to be cleared by writing a 1 to this bit position (Does this mean the state machine stalls?). When this is done, the state machine will drop the current transaction and move on. Writing this bit to 0 has no effect.	WC	0				
6	RESERVED		RO	0	X	X	X	X
5	RESERVED		RO	0	X	X	–	–
5	CS16_ERR	When this bit is set it indicates that a CS16 error has been found by the TDMA.	WC	0	–	–	X	X

Table 316: TDMA Command Register (tdma_command, Offset 0x5c00) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
4	MASTER_ABORT	This bit indicates that the current transaction received a master abort. When this bit is set, the TDMA block will wait for it to be cleared by writing a 1 to this bit position. While in this state, the Host Coalescing block will not free the DMA channel where the error was encountered.	WC	0	X	X	-	-
4	RESERVED		RO	0	-	-	X	X
3-1	RESERVED		RO	0	X	X	X	X
0	ENABLED	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the block is enabled. Writing this bit as a 0 has no effect. This bit is controlled by the "MISC Enable Set Register (misc_enable_set_bits, Offset 0x810)" on page 232.	RO	0	X	X	X	X

TDMA STATUS REGISTER (TDMA_STATUS, OFFSET 0x5c04)**Table 317: TDMA Status Register (tdma_status, Offset 0x5c04)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31-18	RESERVED		RO	0	X	X	-	-
31	IFIFO_OVERFLOW	When this bit is set an input FIFO overflow has occurred.	RO	0	-	-	X	X
30	OFIFO_OVERFLOW	When this bit is set and ouput FIFO overflows has occurred.	RO	0	-	-	X	X
29-26	RESERVED		RO	0	-	-	X	X
25-20	MAX_IFIFO_DEPTH	This field indicates the maximum number of input FIFO addresses used.	RO	0	-	-	X	X
19-18	RESERVED		RO	0	-	-	X	X
17	BURST_CNT	This bit is a generic statistic for the number of DMA transactions that have been processed by the TDMA. This bit toggles once each time a request is made to the DMA block to cause a PCI transfer.	RO	0	X	X	X	X
16	FTQ_ENTRY_CNT	This bit is a generic statistic for the number of FTQ entries that have been processed by the TDMA. This bit toggles once each time a FTQ entry is processed.	RO	0	X	X	X	X
15-4	RESERVED		RO	0	X	X	X	X
3	LOCK_WAIT	This bit indicates that the current record in the FTQ is waiting because the lock was not immediately available to complete the DMA. This is not an error, but could indicate a problem down-stream if set during chip idle state.	RO	0	X	X	X	X

Table 317: TDMA Status Register (tdma_status, Offset 0x5c04) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2	PATCH_FTQ_WAIT	This bit indicates that the current record in the FTQ is waiting because there is not enough room in the TX Patchup FTQ to complete the DMA. This is not an error, but could indicate a problem down-stream if set during chip idle state.	RO	0	X	X	X	X
1	PAYLOAD_WAIT	This bit indicates that the current record in the FTQ is waiting because there is not enough room in the TX Payload Queue to complete the DMA. This is not an error, but could indicate a problem down-stream if set during chip idle state.	RO	0	X	X	X	X
0	DMA_WAIT	This bit indicates that the current record in the FTQ is waiting because there are no available DMA channels. This is not an error, but could indicate a problem down-stream if set during chip idle state.	RO	0	X	X	X	X

TDMA CONFIGURATION REGISTER (TDMA_CONFIG, OFFSET 0x5c08)**Table 318: TDMA Configuration Register (tdma_config, Offset 0x5c08)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	—	—
23–20	FIFO_CMP	This value is a tuning parameter. This values controls the number of words the TX DMA will hold in its internal FIFO before pausing reads from the DMA Engine.	RW	0x9	X	X	—	—
19–18	RESERVED		RO	0	X	X	—	—
17	CMPL_ENTRY	This bit controls the maximum number of TX DMA operation completions that can be queued at one time. When this bit is: <ul style="list-style-type: none"> Clear, three entries can be queued before the TX DMA stalls Set, two entries can be queued 	RW	0	X	—	—	—
17	CMPL_ENTRY	This register is only applicable to revision A1 and later.	RW	0	—	X	—	—
16	CHK_L2_BD	This bit enables a mode where L2 DMA jobs are verified to completely consume the last BD. If this bit is 1 and the error condition occurs, then the BAD_L2_LENGTH_ABORT bit of the “TDMA Command Register (tdma_command, Offset 0x5c00)” on page 457 will be set.	RW	0	X	X	X	X
15	ALIGN_ENA	When this bit is 1, TX DMA jobs of lengths equal or greater than LIMIT_SZ will be aligned to the host address boundary defined by LINE_SZ.	RW	0	X	X	X	X
14–12	RESERVED		RO	0	X	X	X	X

Table 318: TDMA Configuration Register (tdma_config, Offset 0x5c08) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11–8	LINE_SZ	These bits control where the DMA alignment boundary (cache line size) is set. This is the address granularity where DMA jobs larger or equal to the LIMIT_SZ size will be split.	RW	0x8	X	X	X	X
		Value Name Description						
		0 64 DMA jobs are aligned to 64-byte boundaries.						
		4 128 DMA jobs are aligned to 128-byte boundaries.						
		6 256 DMA jobs are aligned to 256-byte boundaries.						
		8 512 DMA jobs are aligned to 512-byte boundaries.						
7–4	LIMIT_SZ	These bits control when the DMA alignment algorithm will be used. DMA jobs that are shorter than this value will not be aligned.	RW	0x8	X	X	X	X
		Value Name Description						
		0 64 DMA jobs less than 64 will not be aligned.						
		0x4 128 DMA jobs less than 128 will not be aligned.						
		0x6 256 DMA jobs less than 256 will not be aligned.						
		0x8 512 DMA jobs less than 512 will not be aligned.						
3–2	RESERVED		RO	0	X	X	X	X
1	ONE_RECORD	When this bit is set, the TX DMA block will limit itself to processing one record at a time and will not look ahead into the FTQ to process multiple records at once. Multiple DMA requests may still be made on the same FTQ record. This is intended as a debug tool only.	RW	0	X	X	X	X
0	ONE_DMA	When this bit is set, the TX DMA block will limit itself to one request to the DMA Block at a time. This setting is intended as a debug tool only.	RW	0	X	X	X	X

TDMA PAYLOAD PRODUCER REGISTER (TDMA_PAYLOAD_PROD, OFFSET 0x5c0c)*Table 319: TDMA Payload Producer Register (tdma_payload_prod, Offset 0x5c0c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–3	VALUE	This is the producer memory Offset into the TX Payload queue that is being used by the TX DMA block. Don't change this value unless everything is stopped and you are updating the corresponding consumer value in the TX Assembler to match. This value points to the next word that will be written by TX DMA. The bottom 3 bits are always 0. The top bit is used as a pass toggle bit for tracking empty/full conditions.	RW	0	X	X	X	X
2–0	RESERVED		RO	0	X	X	X	X

TDMA DMAD FSM REGISTER (TDMA_DMAD_FSM, OFFSET 0x5c80)*Table 320: TDMA DMAD FSM Register (tdma_dmad_fsm, Offset 0x5c80)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	X	X
27–24	BD	BD Read FSM	RO	0	X	X	—	—
27–0	RESERVED		RO	0	—	—	X	X
23	RESERVED		RO	0	X	X	—	—
22–20	DMAD	DMAD FSM	RO	0	X	X	—	—
19–17	RESERVED		RO	0	X	X	—	—
16	DR_INTF	DMA Read interface FSM	RO	0	X	X	—	—
15–13	RESERVED		RO	0	X	X	—	—
12	ARB_CTX	Context Arbitration FSM	RO	0	X	X	—	—
11–10	RESERVED		RO	0	X	X	—	—
9–8	ARB_TBDC	TBDC Arbitration FSM	RO	0	X	X	—	—
7–4	PUSH	Push FSM	RO	0	X	X	—	—
3–1	RESERVED		RO	0	X	X	—	—
0	BD_INVLD	BD Invalidate FSM	RO	0	X	X	—	—

TDMA DMAD STATUS REGISTER (TDMA_DMAD_STATUS, OFFSET 0x5c84)*Table 321: TDMA DMAD Status Register (tdma_dmad_status, Offset 0x5c84)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15–12	IFTQ_ENUM	Input FTQ Entry Number	RO	0	X	X	—	—
15–0	RESERVED		RO	0	—	—	X	X
11–10	RESERVED		RO	0	X	X	—	—
9–8	RHOLD_BD_ENTRY	Number of BD's currently active	RO	0	X	X	—	—
7–6	RESERVED		RO	0	X	X	—	—
5–4	RHOLD_DMAD_ENTRY	Number of DMADs' currently active	RO	0	X	X	—	—
3–2	RESERVED		RO	0	X	X	—	—
1–0	RHOLD_PUSH_ENTRY	Number of Push commands currently active	RO	0	X	X	—	—

TDMA DR INTERFACE FSM REGISTER (TDMA_DR_INTF_FSM, OFFSET 0x5c88)*Table 322: TDMA DR Interface FSM Register (tdma_dr_intf_fsm, Offset 0x5c88)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–19	RESERVED		RO	0	X	X	X	X
18–16	DMAD	DMAD FSM	RO	0	X	X	—	—
15–0	RESERVED		RO	0	—	—	X	X
15	RESERVED		RO	0	X	X	—	—
14–12	DR_BUF	DMA Read buffer interface FSM	RO	0	X	X	—	—



Table 322: TDMA DR Interface FSM Register (*tdma_dr_intf_fsm*, Offset 0x5c88) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11–10	RESERVED		RO	0	X	X	—	—
9–8	TPBUF	TPBUF write FSM	RO	0	X	X	—	—
7	RESERVED		RO	0	X	X	—	—
6–4	TPATQ	TPATQ interface FSM	RO	0	X	X	—	—
3–2	RESERVED		RO	0	X	X	—	—
1–0	L2_COMP	Layer2 Completion FSM	RO	0	X	X	—	—

TDMA DR INTERFACE STATUS REGISTER (TDMA_DR_INTF_STATUS, OFFSET 0x5C8C)**Table 323: TDMA DR Interface Status Register (*tdma_dr_intf_status*, Offset 0x5c8c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–19	RESERVED		RO	0	X	X	X	X
18–16	BYTE_COUNT	Byte Count	RO	0	X	X	—	—
18–0	RESERVED		RO	0	—	—	X	X
15–12	NXT_PNTR	Next Pointer	RO	0	X	X	—	—
11	RESERVED		RO	0	X	X	—	—
10–8	SHIFT_ADDR	Right Shift address—Amount of the data from DMA buffer to be shifted	RO	0	X	X	—	—
7–6	RESERVED		RO	0	X	X	—	—
5–4	DATA_AVAIL	Data Available Select ID—which of the three pipelined operation is currently active.	RO	0	X	X	—	—
3	RESERVED		RO	0	X	X	—	—
2–0	HOLE_PHASE	Hole_Phase—Current Hole Phase	RO	0	X	X	—	—

TDMA FTQ DATA REGISTERS (TDMA_FTQ_DATA[14], OFFSET 0x5FC0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	0		X	X	X	X

TDMA FTQ COMMAND REGISTER (TDMA_FTQ_CMD, OFFSET 0x5FF8)

The TDMA FTQ is 16 records deep.

Table 324: TDMA FTQ Command Register (*tdma_ftq_cmd*, Offset 0x5ff8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X

Table 324: TDMA FTQ Command Register (tdma_ftq_cmd, Offset 0x5ff8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the INTERVENE bit set as indicated in the “ TDMA FTQ Control Register (tdma_ftq_ctl, Offset 0x5fc) ” on page 465. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “ TDMA FTQ Control Register (tdma_ftq_ctl, Offset 0x5fc) ” on page 465. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the “ TDMA FTQ Data Registers (tdma_ftq_data[14], Offset 0x5fc0) ” on page 756 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the current “ TDMA FTQ Data Registers (tdma_ftq_data[14], Offset 0x5fc0) ” on page 756 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the intervene bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “ TDMA FTQ Data Registers (tdma_ftq_data[14], Offset 0x5fc0) ” on page 756 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X

Table 324: TDMA FTQ Command Register (*tdma_ftq_cmd*, Offset 0x5ff8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “TDMA FTQ Data Registers (<i>tdma_ftq_data[14]</i> , Offset 0x5fc0)” on page 756.	RW	0	X	X	X	X
		Val Name Description						
		0 0 When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.						
		1 1 When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an intervene record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.						
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field in the “TDMA FTQ Control Register (<i>tdma_ftq_ctl</i> , Offset 0x5ffc)” on page 465.	RW	0	X	X	X	X

TDMA FTQ CONTROL REGISTER (TDMA_FTQ_CTL, OFFSET 0x5FFC)Table 325: TDMA FTQ Control Register (*tdma_ftq_ctl*, Offset 0x5ffc)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X

Table 325: TDMA FTQ Control Register (*tdma_ftq_ctl*, Offset 0x5ffc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

DEBUG UART (DBU) REGISTERS

The Debug UART (DBU) block provides a means to perform GRC read and write cycles from a remote terminal connected via a serial cable. The DBU supports these operations without the involvement of on-chip firmware or a PCIe connected host PC, allowing access to the NetXtreme II device in any situation where the power is applied to the controller. A state-machine based text debugger interprets and responds to debugger commands.

Debug Port Access

After the debug port is made available on a system design, a driver developer may access internal registers through a serial communications utility such as HyperTerminal for Windows® using the following documented commands. The serial settings for the debug port are 19,200 baud, no parity, eight data bits, one stop bit (or 19,200,N,8,1) with no flow control.

After communication has been established with the NetXtreme II controller, the terminal utility displays a > as the command prompt. If no command prompt is displayed, press the Enter key once or twice until the command prompt is displayed.

The following commands are supported:

- r<addr>—The read command returns the 32-bit value of the register specified in the mandatory <addr> field.
- w<addr> <data>—The write command writes the data from the mandatory <data> field to the register specified in the mandatory <addr> field.
- d[addr]—The dump command reads for 256 registers beginning at the register specified by the optional [addr] field. When the d command is used without an [addr] field, it continues reading registers where the last invocation of the d command ended.

All address values and data are given in hexadecimal format and do not use a leading 0x or a trailing h.

DBU COMMAND REGISTER (DBU_COMMAND, OFFSET 0x6000)

Table 326: DBU Command Register (dbu_command, Offset 0x6000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–3	RESERVED		RO	0	X	X	X	X
2–0	RESERVED		RO	0	X	–	–	–
2	RX_OVERFLOW	When this bit is set a receive overflow has occurred.	WC	0	–	X	X	X
1	RX_ERROR	When this bit is set a byte has been received with a framing error.	WC	0	–	X	X	X
0	ENABLE	The DBU block is always enabled.	RO	0x1	–	X	X	X

DBU STATUS REGISTER (DBU_STATUS, OFFSET 0x6004)

Table 327: DBU Status Register (dbu_status, Offset 0x6004)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–2	RESERVED		RO	0	X	X	X	X
1–0	RESERVED		RO	0	X	–	–	–

Table 327: DBU Status Register (dbu_status, Offset 0x6004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	TXDATA_OCCUPIED	When this bit is set there is data waiting to be transmitted in the VALUE field of the “DBU Transmit Data Register (dbu_txdata, Offset 0x6014)” on page 469.	RO	0	–	X	X	X
0	RXDATA_VALID	When this bit is set there is a valid byte waiting to be read in the VALUE field of the “DBU Receive Data Register (dbu_rxdata, Offset 0x6010)” on page 469.	RO	0	–	X	X	X

DBU CONFIGURATION REGISTER (DBU_CONFIG, OFFSET 0x6008)**Table 328: DBU Configuration Register (dbu_config, Offset 0x6008)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–3	RESERVED		RO	0	X	X	X	X
2–0	RESERVED		RO	0	X	–	–	–
2	CRLF_ENABLE	When this bit is set all line feed characters will be proceeded by a carriage return.	RW	0x1	–	X	X	X
1	DEBUGSM_ENABLE	When this bit is set the debug state machine responds to received characters by performing CRG master transactions and returning received data.	RW	0x1	–	X	X	X
0	TIMING_OVERRIDE	When this bit is set the UART timing will be determined by the values in the DMBU Timing register.	RW	0	–	X	X	X

DBU TIMING REGISTER (DBU_TIMING, OFFSET 0x600c)**Table 329: DBU Timing Register (dbu_timing, Offset 0x600c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	–	–	–
31–16	BIT_INTERVAL	This field sets the number of core clock cycles between bits for both transmit and receive operation. The default value of 0x516 results in 19200 baud operation with CK25 at 25MHz.	RW	0x516	–	X	X	X
15–0	FB_SMPL_OFFSET	This field sets the number of core clock cycles after the falling edge of the rx_data pin that the start bit should be sampled. The default value of 0x28b results in 19200 baud operation with CK25 at 25MHz.	RW	0x28b	–	X	X	X

DBU RECEIVE DATA REGISTER (DBU_RXDATA, OFFSET 0x6010)*Table 330: DBU Receive Data Register (dbu_rxdata, Offset 0x6010)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–9	RESERVED		RO	0	X	X	X	X
8–0	RESERVED		RO	0	X	–	–	–
8	ERROR	When this bit is set it indicates that the data currently in the VALUE field was received in error. This bit is only valid if the RXDATA_VALID bit is set in the “DBU Status Register (dbu_status, Offset 0x6004)” on page 467.	RO	0	–	X	X	X
7–0	VALUE	This field represents the most recently received data byte and is only valid if the RXDATA_VALID bit is set in the “DBU Status Register (dbu_status, Offset 0x6004)” on page 467.	RO	0	–	X	X	X

DBU TRANSMIT DATA REGISTER (DBU_TXDATA, OFFSET 0x6014)*Table 331: DBU Transmit Data Register (dbu_txdata, Offset 0x6014)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–8	RESERVED		RO	0	X	X	X	X
7–0	RESERVED		RO	0	X	–	–	–
7–0	VALUE	This field can be written to transmit a single byte of data on the serial interface. Firmware should poll the TXDATA_OCCUPIED bit of the “DBU Status Register (dbu_status, Offset 0x6004)” on page 467 before writing to this register to make sure that a previously written byte has been transmitted. When read this field will return the last data byte that was written.	RW	0	–	X	X	X

NON-VOLATILE MEMORY (NVM) REGISTERS

The non-volatile memory controller (NVM) block provides software access to the external serial SPI flash memory device. Reads and writes to the flash are performed by reading and writing registers in NVM.

NVM COMMAND REGISTER (NVM_COMMAND, OFFSET 0x6400)

Table 332: NVM Command Register (nvm_command, Offset 0x6400)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–23	RESERVED		RO	0	—	—	X	X
22–20	RESERVED		RO	0	X	X	—	—
22	MODE_256	The 256B page size mode disable bit. A 256-byte page mode has been added to the block. This mode is normally on. The mode helps convert a 264B page Atmel part to act more like a 256B page part. For reads, the controller transparently closes the page after byte location 0xFF and opens the next page. For writes, it is the responsibility of FW or SW to close the page at 0xFF and start a new operation on the next page. When this bit is written as '1' while the FIRST bit is set, the 256B page mode is disabled for the next operation. It is self-clearing when both the LAST bit is set and the DONE bit is asserted. Effects Atmel only. No effect with ST devices. [SHARE]	SC	0	—	—	X	X
21	RD_STATUS	The read status command bit. When set, the Flash controller will read the status register from the external Flash device. [SHARE]	RW	0	—	—	X	X
20	RD_ID	The read ID command bit. When set, the Flash controller will read the ID register from the external Flash device. This is specifically for ST devices. Setting this bit for Atmel devices will give the same results as RD STATUS. [SHARE]	RW	0	—	—	X	X
19	WRSR	Setting this bit causes the Flash interface state machine to generate a wrsr_cmd (0x1, hard-wired) to the Flash device through the SPI interface and set the status register of the Flash device to be written with sr_data. This bit only applies to SST25VF512 parts.	RW	0	X	X	—	—
19–18	RESERVED		RO	0	—	—	X	X

Table 332: NVM Command Register (nvm_command, Offset 0x6400) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
18	EWSR	Setting this bit causes the Flash interface state machine to generate a ewsr_cmd (0x50) to the Flash device through the SPI interface and set the status register of the Flash device to be write-enabled. This bit only applies to SST25VF512 parts.	RW	0	X	X	-	-
17	WRDI	Setting this bit causes the Flash interface state machine to generate a wrdi_cmd to the Flash device through the SPI interface and disable writes. [SHARE]	RW	0	X	X	X	X
16	WREN	Setting this bit causes the Flash interface state machine to generate a wren_cmd to the Flash device through the SPI interface and enable writes. [SHARE]	RW	0	X	X	X	X
15–9	RESERVED		RO	0	X	X	X	X
8	LAST	When this bit is set, the next command sequence will be interpreted as the last one of a burst and any remaining administrative operation will be completed. This means that the buffer will be written to Flash memory if needed on a write. [SHARE]	RW	0	X	X	X	X
7	FIRST	This bit is passed to the SEE_FSM or SPI_FSM if the pass_mode bit is set. [SHARE]	RW	0	X	X	X	X
6	ERASE	Set high to execute an erase. This bit is ignored if the WR bit is clear. [SHARE]	RW	0	X	X	X	X
5	WR	Set high to execute write or erase. [SHARE]	RW	0	X	X	X	X
4	DOIT	Command from software to start the defined command. The done bit must be clear before setting this bit. This bit is self-clearing and will remain set while the command is active. [SHARE]	SC	0	X	X	X	X
3	DONE	Sequence completion bit that is asserted when the command requested by assertion of the DOIT bit has completed. Will be cleared while the command is in progress. Will stay asserted until DOIT is reasserted or the DONE bit is cleared by writing a 1 to the DONE bit. The DONE bit is the FLSH_ATTN signal. [SHARE]	WC	0	X	X	X	X
2–1	RESERVED		RO	0	X	X	X	X
0	RST	When set, the entire NVM state machine is reset. This bit is self-clearing. [SHARE]	SC	0	X	X	X	X

NVM WRITE VALUE REGISTER (NVM_WRITE, OFFSET 0x6408)**Table 333: NVM Write Value Register (nvm_write, Offset 0x6408)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	NVM_WRITE_VALUE	32 bits of write data use when write commands are executed. When bitbang_mode is set, only the bottom 6 bits have any effect as described below. [SHARE]	RW	0	–	–	X	X
Value Name Description								
0	BIT_BANG	When in bit-bang mode, the bottom 6 bits of this register control the output value of the external pins.						
Value Name Description								
1	SI	When in bit-bang mode, this bit controls the SI output value.						
2	SO	When in bit-bang mode, this bit controls the SO output value.						
4	CS_B	When in bit-bang mode, this bit controls the CS_B output value.						
8	SCLK	When in bit-bang mode, this bit controls the SCLK output value.						
Value Name Description								
0	BIT_BANG	When in bit-bang mode, the bottom 6 bits of this register control the output value of the external pins.	RW	0	X	X	–	–
1	EECLK	When in bit-bang mode, this bit controls the EECLK output value.						
2	EEDATA	When in bit-bang mode, this bit controls the EEDATA output value.						
4	SCLK	When in bit-bang mode, this bit controls the SCLK output value.						
8	CS_B	When in bit-bang mode, this bit controls the CS_B output value.						
16	SO	When in bit-bang mode, this bit controls the SO output value.						
32	SI	When in bit-bang mode, this bit controls the SI output value.						

NVM ADDRESS REGISTER (NVM_ADDR, OFFSET 0x640c)*Table 334: NVM Address Register (nvm_addr, Offset 0x640c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–24	NVM_ADDR_VALUE	The 24-bit address value used in read, write and erase operations. When in bit-bang mode, the bottom 6 bits control the output enable for each pin. [SHARE]	RO	0	–	–	X	X
Value Name								
0	BIT_BANG	When in bit-bang mode, the bottom 6 bits of this register control the output value of the external pins.						
1	SI	When in bit-bang mode, this bit controls the SI output value.						
2	SO	When in bit-bang mode, this bit controls the SO output value.						
4	CS_B	When in bit-bang mode, this bit controls the CS_B output value.						
8	SCLK	When in bit-bang mode, this bit controls the SCLK output value.						

Table 334: NVM Address Register (*nvm_addr*, Offset 0x640c) (Cont.)

Bit	Name	Description		Mode	Reset	06	08	09	16
23–0 (cont.)	NVM_ADDR_VALUE	Value Name	Description			0	X	X	–
		0 BIT_BANG	When in bit-bang mode, the bottom 6 bits of this register control the output enable value of the external pins.						
		1 EECLK	When in bit-bang mode, this bit controls the enables the EECLK pin as an output when 1.						
		2 EEDATA	When in bit-bang mode, this bit controls the enables the EEDATA pin as an output when 1.						
		4 SCLK	When in bit-bang mode, this bit controls the enables the SCLK pin as an output when 0.						
		8 CS_B	When in bit-bang mode, this bit controls the enables the CS_B pin as an output when 0.						
		16 SO	When in bit-bang mode, this bit controls the enables the SO pin as an output when 0.						
		32 SI	When in bit-bang mode, this bit controls the enables the SI pin as an output when 0.						

NVM READ VALUE REGISTER (NVM_READ, OFFSET 0x6410)*Table 335: NVM Read Value Register (nvm_read, Offset 0x6410)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	NVM_READ_VALUE	The 32 bits of read data returned when read commands are executed. When bitbang_mode is set, bits 0 to 5 reflect the current input value of the external pins. [SHARE]	RO	0	–	–	X	X
Value Name								
0	BIT_BAN G	When in bit-bang mode, the bottom 6 bits of this register control the output value of the external pins.						
1	SI	When in bit-bang mode, this bit controls the SI output value.						
2	SO	When in bit-bang mode, this bit controls the SO output value.						
4	CS_B	When in bit-bang mode, this bit controls the CS_B output value.						
8	SCLK	When in bit-bang mode, this bit controls the SCLK output value.						
Value Name								
0	BIT_BAN G	When in bit-bang mode, the bottom 6 bits of this return the input value of the external pins.	RO	0	X	X	–	–
1	EECLK	When in bit-bang mode, this bit returns the current value of the EECLK pin.						
2	EEDATA	When in bit-bang mode, this bit returns the current value of the EEDATA pin.						
4	SCLK	When in bit-bang mode, this bit returns the current value of the SCLK pin.						
8	CS_B	When in bit-bang mode, this bit returns the current value of the CS_B pin.						
16	SO	When in bit-bang mode, this bit returns the current value of the SO pin.						
32	SI	When in bit-bang mode, this bit returns the current value of the SI pin.						

NVM CONFIGURATION 1 REGISTER (NVM_CFG1, OFFSET 0x6414)*Table 336: NVM Configuration 1 Register (nvm_cfg1, Offset 0x6414)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31	COMPAT_BYPASS_S	Legacy bit. [SHARE]	RW	0	X	X	X	X
30	FW_FLASH_TYPE_EN	This register has no hardware function but can be modified by firmware.	RW	0	X	X	-	-
30	FW_FLASH_TYPE_EN	This bit returns 1 to indicate that flash has already been configured. [SHARE]	RO	0x1	-	-	X	X
29–26	FW_FLASH_TYPE	This register has no hardware function but can be modified by firmware.	RW	0	X	-	-	-
29	FW_USTRAP_3	This bit has no hardware function but can be modified by firmware. This is bit 3 of the “upper” strap value and reflects the actual pin strapped value from the SO pin at core reset.	RW	X	-	X	-	-
29	FW_USTRAP_3	Legacy strap value bit 3. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]	RO	X	-	-	X	X
28	FW_USTRAP_2	This bit has no hardware function but can be modified by firmware. This is bit 2 of the “upper” strap value and reflects the actual pin strapped value from the SI pin at core reset.	RW	X	-	X	-	-
28	FW_USTRAP_2	Legacy strap value bit 2. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]	RO	X	-	-	X	X
27	FW_USTRAP_0	This bit has no hardware function but can be modified by firmware. This is bit 0 of the “upper” strap value and reflects the actual pin strapped value from the SCLK pin at core reset.	RW	X	-	X	-	-
27	FW_USTRAP_0	Legacy strap value bit 0. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]	RO	X	-	-	X	X
26	FW_USTRAP_1	This bit has no hardware function but can be modified by firmware. This is bit 1 of the “upper” strap value and reflects the actual pin strapped value from the CS_B pin at core reset.			-	X	-	-
26	FW_USTRAP_1	Legacy strap value bit 1. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]	RO	X	-	-	X	X
25	FLASH_SIZE	Enables 1-Mbit devices as opposed to 512 Kbit. At CORE reset, this pin is set to the value of the SO pin.	RW	X	X	X	-	-
25	FLASH_SIZE	Legacy strap control bit 3. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]	RW	X	-	-	X	X
24	PROTECT_MODE	Enable protect function when ST M25P10-A is used. At CORE reset, this pin is set to the value of the SI pin.	RW	X	X	X	-	-
24	PROTECT_MODE	Legacy strap control bit 2. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]	RW	X	-	-	X	X
23–22	RESERVED		RO	0	X	-	-	-

Table 336: NVM Configuration 1 Register (nvm_cfg1, Offset 0x6414) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
23	STRAP_CONTROL_0	Lower bit of the STRAP_CONTROL value. This bit is set for "new" devices and clear for legacy devices.			-	B	-	-
					0			
					+			
23	STRAP_CONTROL_0	Legacy strap control bit 1. Set based on new strap values to indicate either Atmel or ST flash. [SHARE]			-	-	X	X
22	RESERVED		RO	0	-	X	X	X
21–11	SEE_CLK_DIV	Divisor used to create all 1x time for all EEPROM Interface I/O pin timing definition A divisor of 0 means that an SCL will transition at minimum of each CORE_CLK rising edge. The equation to calculate the clock frequency for SCL is: CORE_CLK frequency / ((see_clk_div +1) *4)	RW	0x60	X	X	-	-
		Note: SCL is four times slower than 1x time. The reset value of 44 selects is 370 kHz.						
21–11	SEE_CLK_DIV	Legacy value. [SHARE]	RO	0x60	-	-	X	X
10–7	SPI_CLK_DIV	Divisor used to create all 1x time for all Flash Interface I/O pin timing definition A divisor of 0 means that an EECLK will transition at minimum of each CORE_CLK rising edge. The equation to calculate the clock frequency for EECLK is: CORE_CLK frequency / ((spi_clk_div +1) *2). [SHARE]	RW	0x4	X	X	X	X
		Note: EECLK is two times slower than 1x time. The reset value of 4 specifies 6.6 MHz.						
6–4	STATUS_BIT	Bit Offset in status command response to interpret as the ready flag. [SHARE]	RW	0	X	X	X	X
		Value Name	Description					
		0 FLASH_RDY	Ready if FLASH_MODE is 1.					
		7 BUFFER_RDY	Ready if BUFFER_MODE is 1.					
3	BITBANG_MODE	Enable bit-bang mode to control pins. [SHARE]	RW	0	X	X	X	X
2	PASS_MODE	Enable pass-thorough mode to the byte level SPI and SEE state machines. [SHARE]	RW	0	X	X	X	X
1	BUFFER_MODE	Enable SSRAM Buffered Flash Mode when AT45DB is used. At CORE reset, this pin is set to the value of the SCLK pin. [SHARE]	RW	X	X	X	X	X
0	FLASH_MODE	Enable Flash Interface Mode. At CORE reset, this pin is set to the value of the CS_B pin. [SHARE]	RW	X	X	X	X	X

NVM CONFIGURATION 2 REGISTER (NVM_CFG2, OFFSET 0x6418)*Table 337: NVM Configuration 2 Register (nvm_cfg2, Offset 0x6418)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	–	–
31–24	READ_ID	Flash Read ID register command. This command is used to read the ID register from ST devices. Reset value depends on strap values. (0x57 for Atmel; 0x9f for ST). [SHARE]	RW	0	–	–	X	X
23–16	STATUS_CMD	Flash status command. This command is used to poll the ready status of the flash part after many of the commands. Reset value is 0x9Fh if flash_mode=1, 0x57h if buffer_mode=1, and 0x5h if protect_mode =1. [SHARE]	RW	X	X	X	X	X
15–8	DUMMY	Value for dummy bytes added to commands.	RW	X	X	X	–	–
15–8	RESERVED		RO	X	–	–	X	X
7–0	ERASE_CMD	Flash block erase command. ready status will be polled for after this command. Reset value is 0x20h if flash_mode=1, 0x81h if buffer_mode=1, and 0xd8h if protect_mode=1. [SHARE]	RW	X	X	X	X	X

NVM CONFIGURATION 3 REGISTER (NVM_CFG3, OFFSET 0x641C)*Table 338: NVM Configuration 3 Register (nvm_cfg3, Offset 0x641c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	READ_CMD	This is the Flash/SEEPROM read command. Following this command, any number of bytes may be read up to the end of the Flash memory. For SEEPROM (flash_mode=0), this is SEEPROM read command. Bit[26–25] is address bit A1 and A0 of SEEPROM. The two bits base on the value of A1 and A0 assigned to this SEEPROM device should be modified. Reset value is 0xFF if flash_mode=1, 0x68 if buffer_mode=1, 0x3 if protect_mode=1, and 0xA1 if otherwise. [SHARE]	RW	X	X	X	X	X
23–16	BUFFER_WRITE_CMD	If buffer mode is being used, then this command will be executed at the end of a complete write operation. Reset value is 0x84 if buffer_mode=1.	RW	X	X	X	–	–
23–16	RESERVED		RO	X	–	–	X	X

Table 338: NVM Configuration 3 Register (nvm_cfg3, Offset 0x641c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15–8	WRITE_CMD	Command to write one byte to the Flash array or SSRAM buffer, depending on the value of buffer_mode. If BUFFER_MODE is not active, then this command will poll for ready status when complete. For SEEPPROM (flash_mode=0), this is SEEPPROM write command. Bit[10–9] is address bit A1 and A0 of SEEPPROM. The two bits base on the value of A1 and A0 assigned to this SEEPPROM device should be modified. Reset value is 0x10 if flash_mode=1, 0x83 if buffer_mode=1, and 0x2 if protect_mode=1, 0xA0 if otherwise. [SHARE]	RW	X	X	X	X	X
7–0	BUFFER_RD_CMD	Command to transfer Flash value to buffer. This command is executed before the first write command to a new page after the erase command has been executed. Reset value is 0x53h if buffer_mode=1. For Atmel devices, this command is issued automatically upon the "FIRST" write. It is not issued for ST devices since they automatically do this operation internally. [SHARE]	RW	X	X	X	X	X

NVM SOFTWARE ARBITRATION REGISTER (NVM_SW_ARB, OFFSET 0x6420)**Table 339: NVM Software Arbitration Register (nvm_sw_arb, Offset 0x6420)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	REQ3	This is the status of requester 3. When this bit is one, it means that REQ_SET3 has been set since REQ_CLR3. [SPLIT]	RO	0	X	X	X	X
14	REQ2	This is the status of requester 2. When this bit is one, it means that REQ_SET2 has been set since REQ_CLR2.	RO	0	X	X	X	X
13	REQ1	This is the status of requester 1. When this bit is one, it means that REQ_SET1 has been set since REQ_CLR1.	RO	0	X	X	X	X
12	REQ0	This is the status of requester 0. When this bit is one, it means that REQ_SET0 has been set since REQ_CLR0.	RO	0	X	X	X	X

Table 339: NVM Software Arbitration Register (nvm_sw_arb, Offset 0x6420) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
11	ARB_ARB3	When REQ1 arbitration is won, this bit will be read as 1, when an operation is complete, then the CLR_ARB3 must be written to clear this bit.	RO	0	X	X	X	X
10	ARB_ARB2	When REQ1 arbitration is won, this bit will be read as 1, when an operation is complete, then the CLR_ARB2 must be written to clear this bit.	RO	0	X	X	X	X
9	ARB_ARB1	When REQ1 arbitration is won, this bit will be read as 1, when an operation is complete, then the CLR_ARB1 must be written to clear this bit.	RO	0	X	X	X	X
8	ARB_ARB0	When REQ0 arbitration is won, this bit will be read as 1, when an operation is complete, then the CLR_ARB0 must be written to clear this bit. At that point, the next Arb bit will read as 1. At any time, only one of the ARB[3–0] bits will be read as a 1. Arb0 has highest priority, and Arb3 has lowest priority.	RO	0	X	X	X	X
7	ARB_REQ_CLR3	Write this bit as a 1 to clear req3 bit.	SC	0	X	X	X	X
6	ARB_REQ_CLR2	Write this bit as a 1 to clear req2 bit.	SC	0	X	X	X	X
5	ARB_REQ_CLR1	Write this bit as a 1 to clear req1 bit.	SC	0	X	X	X	X
4	ARB_REQ_CLR0	Write this bit as a 1 to clear req0 bit.	SC	0	X	X	X	X
3	ARB_REQ_SET3	Set Software Arbitration request Bit 3. This bit is set by writing a 1 to this bit position.	SC	0	X	X	X	X
2	ARB_REQ_SET2	Set Software Arbitration request Bit 2. This bit is set by writing a 1 to this bit position.	SC	0	X	X	X	X
1	ARB_REQ_SET1	Set Software Arbitration request Bit 1. This bit is set by writing a 1 to this bit position.	SC	0	X	X	X	X
0	ARB_REQ_SET0	Set Software Arbitration request Bit 0. This bit is set by writing a 1 to this bit position.	SC	0	X	X	X	X

NVM ACCESS ENABLE REGISTER (NVM_ACCESS_ENABLE, OFFSET 0x6424)**Table 340: NVM Access Enable Register (nvm_access_enable, Offset 0x6424)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–2	RESERVED		RO	0	X	X	X	X
1	WR_EN	Set this bit 0 will mask the WR command in command register if mode control register bit 21 is 0. Set this bit 1 will allow WR command in command register to be issued even if bit 21 of mode control register is 0. [SHARE]	RW	0	X	X	X	X

Table 340: NVM Access Enable Register (nvm_access_enable, Offset 0x6424) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	EN	Write 1 to this bit will allow write to all the other registers except software arbitration register in this block. Write 0 will prevent those registers from the write. [SHARE]	RW	0	X	X	X	X

NVM WRITE 1 REGISTER (NVM_WRITE1, OFFSET 0x6428)**Table 341: NVM Write 1 Register (nvm_write1, Offset 0x6428)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–16	SR_DATA	Data written into status register for SST25VF512.	RW	0	X	X	—	—
23–16	RESERVED		RO	0	—	—	X	X
15–8	WRDI_CMD	Flash write disable command when device with protection function is used. This command will be issued by the Flash interface state machine through SPI interface to Flash device, and make the Flash device write-disabled. [SHARE]	RW	0	X	X	X	X
7–0	WREN_CMD	Flash write enable command when device with protection function is used. This command will be issued by the Flash interface state machine through SPI interface to Flash device, and make the Flash device write-enabled. [SHARE]	RW	0	X	X	X	X

NVM CONFIGURATION FOUR REGISTER (NVM_CFG4, OFFSET 0x642c)**Table 342: NVM Configuration Four Register (NVM_CFG4, Offset 0x642c—BCM5709 Only)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31–7	RESERVED		RW	0	—	—	X	X
6	STATUS_BIT_POLARITY	This bit determines how the status bit of the device status register is interpreted by hardware. If 0, then 0 means "ready". If 1, then 1 means "ready". For Atmel, this defaults to 1. For ST, this defaults to 0. This value is self-configured on reset based on the strap values. It can be overridden.	RW	X	—	—	X	X

Table 342: NVM Configuration Four Register (NVM_CFG4, Offset 0x642c—BCM5709 Only) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16																											
5–4	MODE_256_EMPTY_BIT _LOC	Bit location for hardware to insert an empty address bit when MODE 256 is not set with Atmel devices. This value is self-configured on reset based on the external device. Note: Max Atmel device size is 64 Mbit due to different bus protocols.	RW	X	—	—	X	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>BIT8</td><td>For 264-Byte pages</td></tr> <tr> <td>1</td><td>BIT9</td><td>For 528-Byte pages</td></tr> <tr> <td>2</td><td>BIT10</td><td>For 1056-Byte pages</td></tr> <tr> <td>3</td><td>BIT11</td><td>For 2112-Byte pages</td></tr> </tbody> </table>	Value	Name	Description	0	BIT8	For 264-Byte pages	1	BIT9	For 528-Byte pages	2	BIT10	For 1056-Byte pages	3	BIT11	For 2112-Byte pages																		
Value	Name	Description																																	
0	BIT8	For 264-Byte pages																																	
1	BIT9	For 528-Byte pages																																	
2	BIT10	For 1056-Byte pages																																	
3	BIT11	For 2112-Byte pages																																	
3	FLASH_VENDOR	This bit is self-configured on reset based on the strap values. It can be overridden.	RW	X	—	—	X	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>ST</td><td>—</td></tr> <tr> <td>1</td><td>ATMEL</td><td>—</td></tr> </tbody> </table>	Value	Name	Description	0	ST	—	1	ATMEL	—																								
Value	Name	Description																																	
0	ST	—																																	
1	ATMEL	—																																	
2–0	FLASH_SIZE	Size of the external Flash device. This information is not used by FLSH hardware. It is only used by software. This value is self-configured on reset based on the external device.	RW	X	—	—	X	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>1MBIT</td><td>—</td></tr> <tr> <td>1</td><td>2MBIT</td><td>—</td></tr> <tr> <td>2</td><td>4MBIT</td><td>—</td></tr> <tr> <td>3</td><td>8MBIT</td><td>—</td></tr> <tr> <td>4</td><td>16MBIT</td><td>—</td></tr> <tr> <td>5</td><td>32MBIT</td><td>—</td></tr> <tr> <td>6</td><td>64MBIT</td><td>—</td></tr> <tr> <td>7</td><td>128MBIT</td><td>—</td></tr> </tbody> </table>	Value	Name	Description	0	1MBIT	—	1	2MBIT	—	2	4MBIT	—	3	8MBIT	—	4	16MBIT	—	5	32MBIT	—	6	64MBIT	—	7	128MBIT	—						
Value	Name	Description																																	
0	1MBIT	—																																	
1	2MBIT	—																																	
2	4MBIT	—																																	
3	8MBIT	—																																	
4	16MBIT	—																																	
5	32MBIT	—																																	
6	64MBIT	—																																	
7	128MBIT	—																																	

NVM RECONFIGURATION REGISTER (NVM_RECONFIG, OFFSET 0x6430)**Table 343: NVM Reconfiguration Register (nvm_reconfig, Offset 0x6430—BCM5709 Only)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	X	—	—
31	RECONFIG_DONE	This bit is 0 on reset. After software finishes reconfiguring FLSH, they will set this bit to 1 to indicate that FLSH has been reconfigured. This bit has no hardware functionality.	RW	0	—	—		
30–8	RESERVED	Reserved for future use.	RW	0	—	—		

Table 343: NVM Reconfiguration Register (nvm_reconfig, Offset 0x6430—BCM5709 Only) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7–4	RECONFIG_STRAP_VAL UE	Used by software to numerically encode how the FLSH has been reconfigured. On reset, this register is set to the same value as DRIG STRAP VALUE. These bits have no hardware functionality.	RW	X	–	–	X	X
3–0	ORIG_STRAP_VALUE	Strap value from I/O logic pins. Only bit[0] is used. Bits[3:1] are for future support.	RO	X	–	–	X	X
Value Name Description								
0	ST	–						
1	ATMEL	–						

HOST COALESCING (HC) REGISTERS

The Host Coalescing (HC) block writes status blocks to host memory and generates interrupts, writes statistics data to host memory, and maintains generic statistics counters.

HC COMMAND REGISTER (HC_COMMAND, OFFSET 0x6800)

Table 344: HC Command Register (hc_command, Offset 0x6800)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–28	RESERVED		RO	0	X	X	X	X
27–22	RESERVED		RO	0	X	—	—	—
27–23	RESERVED		RO	0	—	X	—	—
27	COAL_ON_NEXT_EVENT	When this bit is set a status block will be generated on the next event. This is similar to the COAL_NOW bit but is intended for backwards compatibility with older drivers.	RW	0	—	—	X	X
26–22	RESERVED		RO	0	—	—	X	X
22	MAIN_PWR_INT	This bit is set each time PERST_B falls.	WC	0	—	B 0 +	—	—
21	CLR_STAT_NOW	When this bit is set the statistics RAM is cleared, regardless of statistic mode or block enable settings. During the RAM clear operation, memory parity output must be disabled. When the command is complete, the bit clears back to 0. Writing this bit to a 0 has no effect.	SC	0	X	X	X	X
20–19	FORCE_INT	These bits allow the interrupt output to be controlled manually for testing or debug purposes. These bits are not intended for use in normal operation.	RW	0x3	X	X	X	X
Value	Name	Description						
0	NUL_L	Writing the value 0x0 has no effect on the interrupt. Any forcing remains in effect. Because these bits always read as 0, other threads can modify other command bits without effecting interrupt forcing. Writing this value does not affect the read value of these bits.						
1	HIG_H	Writing the value 0x1 forces the interrupt high until either 2 or 3 are written.						
2	LO_W	Writing the value 0x2 forces the interrupt low until either 1 or 3 are written.						
3	FRE_E	Writing the value 0x1 allows the interrupt to work normally again.						

Table 344: HC Command Register (hc_command, Offset 0x6800) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
18	STATS_NOW	Writing this bit as 1 forces a statistics block to be generated as quickly as possible, regardless of all HC settings. When the command is completed, this bit is cleared to 0 again. Writing this bit to a 0 has no effect.	SC	0	X	X	X	X
17	COAL_NOW_WO_INT	This bit is identical to the COAL_NOW bit, except that no interrupt is generated after the status block has been generated. This bit may be used by a driver that wants to poll some elements of the status block but continue to use interrupts for other elements.	SC	0	X	X	X	X
16	COAL_NOW	Writing this bit as 1 forces a status block to be generated as quickly as possible, regardless of any other HC settings. When the command is completed, this bit is cleared to 0 again. Writing this bit to a 0 has no effect.	SC	0	X	X	X	X
15–5	RESERVED		RO	0	X	X	X	X
4	SKIP_ABORT	This bit always reads as 0. When written as a 1, the HC state machine drops the current transaction and moves on. It is only valid to set this bit when the HC block has set the MASTER_ABORT bit of the "HC Status Register (hc_status, Offset 0x6804)" on page 485. Writing this bit to 0 has no effect.	SC	0	—	—	X	X
4–1	RESERVED		RO	0	—	—	X	X
3–1	RESERVED		RO	0	X	X	—	—
0	ENABLE	This bit must be set to enable the TX Event Read block. This must be done after the ring address and size is programmed. This bit is cleared only by the core-reset process.	RO	0	X	X	X	X

HC STATUS REGISTER (HC_STATUS, OFFSET 0x6804)

Table 345: HC Status Register (hc_status, Offset 0x6804)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25	CORE_CLKS_DURING_SW_IN_TACK_STAT	This bit is the generic statistic bit for the number of core clock counts that have occurred between the first time the interrupt mailbox is written after an interrupt is generated and the last time the interrupt mailbox was written. The last mailbox write is defined as the write that takes bit 0 from 1 to 0. This bit toggles state each time 64 core clocks have elapsed in this state.	RO	0	X	X	X	X

Table 345: HC Status Register (hc_status, Offset 0x6804) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
24	CORE_CLKS_TO_SW_INTACK_STAT	This bit is the generic statistic bit for the number of core clocks that have occurred between the time the interrupt mailbox is written the first time after an interrupt is generated and the second time the interrupt mailbox is written. This bit toggles state each time 64 core clocks have elapsed in this state.	RO	0	X	X	X	X
23	CORE_CLKS_TO_HW_INTACK_STAT	This bit is the generic statistics bit for the number of core clocks that have occurred between the assertion of the INTA output and when the interrupt mailbox is written the first time. The first mailbox write is defined as the write that takes bit 0 from 0 to 1. This bit toggles state each time 64 core clocks have elapsed in this state.	RO	0	X	X	X	X
22-21	RESERVED		RO	0	X	X	X	X
20	NUM_INT_MBOX_WR_STAT	This bit is the generic statistics bit for the number of times the interrupt mailbox has been written. It toggles each time the interrupt mailbox is written.	RO	0	X	X	X	X
19	NUM_INT_GEN_STAT	This bit is the generic statistics bit for the number of times the interrupt line INTA has been asserted. It toggles each time the INTA output is asserted.	RO	0	X	X	X	X
18	NUM_STATUS_BLOCKS_STAT	This bit is the generic statistics bit for the number of status blocks generated. It toggles each time a status block is generated.	RO	0	X	X	X	X
17	CORE_CLK_CNT_STAT	This bit is the generic statistics bit for the number of CORE Clocks. It will toggle once for every 64 CORE clocks that have occurred.	RO	0	X	X	X	X
16	PCI_CLK_CNT_STAT	This bit is the generic statistics bit for the number of PCI Clocks. It will toggle once for every 64 PCI clocks that have occurred.	RO	0	X	X	-	-
16	RESERVED		RO	0	-	-	X	X
15-2	RESERVED		RO	0	X	X	X	X
1	PARITY_ERROR_STATE	This bit indicates that the HC block has detected that the parity error condition exists and it is preventing status block index values from incrementing and generating status blocks based only on the new parity error attention status.	RO	0	X	X	X	X

Table 345: HC Status Register (hc_status, Offset 0x6804) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	MASTER_ABORT	This bit indicates that the current transaction received a master abort. When this bit is set, the HC block will wait for the SKIP_ABORT bit of the "HC Command Register (hc_command, Offset 0x6800)" on page 484 to be written as a 1. While in this state, the HC block will not free the DMA channel where the error was encountered.	RO	0	X	X	-	-
0	RESERVED		RO	0	-	-	X	X

HC CONFIGURATION REGISTER (HC_CONFIG, OFFSET 0x6808)**Table 346: HC Configuration Register (hc_config, Offset 0x6808)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	-	-
31	TX_SEL	When this bit is set the HC uses the TPAT interface for TX updates, otherwise the TDMA interface is used.	RW	0	-	-	X	X
30	UNMASK_ALL	The HC block maintains a single set of mask bits for each interrupt vector which can be updated through either the MSI-X mask field or the INTERRUPT_NUM field of the "Interrupt Acknowledge Command Register (pcicfg_int_ack_cmd, Offset 0x84)" on page 177 . Since MSI-X requires the mask to be enabled after reset, the device driver has two alternatives. It can individually unmask the interrupt vector using the INTERRUPT_NUM field or set this bit to unmask all vectors.	SC	0	-	-	X	X
29	GEN_STAT_AVG_INTR	When this bit is set the generic statistic counters HC_CORE_CLKS_TO_HW_INTACK, HC_CORE_CLKS_TO_SW_INTACK, and HC_CORE_CLKS_DURING_SW_IN TACK, will be maintained across multiple interrupts. The default is to clear the appropriate internal tick generation counter bits for each new interrupt.	RW	0	-	-	X	X
28–27	RESERVED		RO	0	-	-	X	X
26–24	SB_ADDR_INC	This field sets the host address offsets for the status blocks and sets the stride between the status block and each per-processor status block.	RW	0	-	-	X	X

Table 346: HC Configuration Register (hc_config, Offset 0x6808) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
23–20	PER_COLLECT_LIMIT	This field controls the number of status blocks that can be “stored” when the periodic collect feature is enabled. This limits the counter maximum that increments each time the periodic collect time expires with no status block generated. Each count is a credit that can be used to generate more than one interrupt in a later periodic collect timer period.	RW	0xf	—	—	X	X
19	SET_MASK_AT_RD	When this bit is set the interrupt mask will be set automatically when the 0x6848 register is read for every status block that was reported as dirty. This allows the ISR to read the 0x6848 register and directly process the indicated status blocks without having to write to the “ Interrupt Acknowledge Command Register (pcifcfg_int_ack_cmd, Offset 0x84) ” on page 177.	RW	0x1	—	—	X	X
18	USE_INT_PARAM	When this bit is set the HC will automatically switch to using the interrupt mode parameters for the default status block when an MSI/MSI-X message is generated or in legacy mode when the 0x6848 register is read and the default status block interrupt is masked.	RW	0	—	—	X	X
17	ONE_SHOT	When this bit is set the HC will automatically set the INT_MASK bit locally after a MSI/MSI-X message is generated.	RW	0	—	—	X	X
16	PER_MODE	When this bit is set PERIODIC_COLLECT mode is selected which causes a status block to be generated after the collect timer expires following a periodic event. When this bit is clear PERIODIC mode is selected.	RW	0	—	—	X	X
15–8	STAT_MEM_ADDR	This field sets the address for debug read-write access of the statistics accumulation memory. No access is taken when this value is written.	RW	0	X	X	X	X
7	RESERVED		RO	0	X	X	X	X
6	STATUS_PRIORITY	This bit sets the priority used for the DMA channel request for status blocks. A value of 1 sets high priority.	RW	0x1	X	X	X	X
5	STATISTIC_PRIORITY	This bit sets the priority used for the DMA channel request for statistics blocks. A value of 1 sets high priority.	RW	0	X	X	X	X

Table 346: HC Configuration Register (hc_config, Offset 0x6808) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
4	CMD_TMR_MODE	This bit controls the Command Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	X	X	X	X
3	COM_TMR_MODE	This bit controls the Completion Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	X	X	X	X
2	TX_TMR_MODE	This bit controls the TX Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	X	X	X	X
1	RX_TMR_MODE	This bit controls the RX Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	X	X	X	X
0	COLLECT_STATS	This bit enables the collection of statistics information from inside the chip over the register bus. Results are accumulated into the local statistics memory. For collection to be performed this bit must be set to 1 and the "HC Statistics Collect Ticks Register (hc_stat_collect_ticks, Offset 0x6840)" on page 495 must be greater than 0.	RW	0x1	X	X	X	X

HC ATTENTION BITS ENABLE REGISTER (HC_ATTN_BITS_ENABLE, OFFSET 0x680c)

Description	Mode	Reset	06	08	09	16
This register provides a bit-for-bit enable capability of the status_attn_bits field of the RW "Host Status Block (Status)" on page 73. If the corresponding bit in this register is set to 1, the input from the chip is passed through to the status block attn-bits detection logic. If the corresponding bit is left as 0 (the reset state), the corresponding attn-bit is always 0 in the status block.	0	X X X X				

HC STATUS ADDRESS Low REGISTER (HC_STATUS_ADDR_L, OFFSET 0x6810)

Description	Mode	Reset	06	08	09	16
This register controls the lower half of the start address in host physical memory that RW the status block will be placed.	0	X X X X				

HC STATUS ADDRESS HIGH REGISTER (HC_STATUS_ADDR_H, OFFSET 0x6814)

Description	Mode	Reset	06	08	09	16
This register controls the upper half of the start address in host physical memory that RW the status block will be placed.	0	X X X X				

HC STATISTICS ADDRESS Low REGISTER (HC_STATISTICS_ADDR_L, OFFSET 0x6818)

Description	Mode	Reset	06	08	09	16
This register controls the lower half of the start address in host physical memory that RW the statistics block will be placed.	0	X X X X				

HC STATISTICS ADDRESS HIGH REGISTER (HC_STATISTICS_ADDR_H, OFFSET 0x681C)

Description	Mode	Reset	06	08	09	16
This register controls the upper half of the start address in host physical memory that RW the statistics block will be placed.	0	X X X X				

HC TX QUICK CONSUMER TRIP POINTER REGISTER (HC_TX_QUICK_CONS_TRIP, OFFSET 0x6820)

Table 347: HC TX Quick Consumer Trip Pointer Register (hc_tx_quick_cons_trip, Offset 0x6820)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–16	INT	This value controls the number of TX Quick BD Chain entries that must be completed before a status block is generated during an interrupt.	RW	0x1	X	X	X	X
15–8	RESERVED		RO	0	X	X	X	X



Table 347: HC TX Quick Consumer Trip Pointer Register (hc_tx_quick_cons_trip, Offset 0x6820) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7–0	VALUE	This value controls the number of TX Quick BD Chain entries that must be completed before a status block is generated. Setting this to 0 disables TX Quick BD Chain consumption from generating status blocks.	RW	0x1	X	X	X	X

HC KERNEL COMPLETION PRODUCER TRIP POINT REGISTER (HC_COMP_PROD_TRIP, OFFSET 0x6824)

Table 348: HC Kernel Completion Producer Trip Point Register (hc_comp_prod_trip, Offset 0x6824)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–16	INT	This value controls the number of Kernel/Command Completion Chain entries that must be written before a status block is generated during an interrupt.	RW	0x1	X	X	X	X
15–8	RESERVED		RO	0	X	X	X	X
7–0	VALUE	This value controls the number of Kernel or Command Completion Chain entries that must be written before a status block is generated. Setting this to 0 disables Completion production from generating status blocks.	RW	0x1	X	X	X	X

HC RX QUICK CONSUMER TRIP POINT REGISTER (HC_RX_QUICK_CONS_TRIP, OFFSET 0x6828)

Table 349: HC RX Quick Consumer Trip Point Register (hc_rx_quick_cons_trip, Offset 0x6828)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–16	INT	This value controls the number of RX Quick BD entries that must be completed before a status block is generated during interrupt processing.	RW	0x1	X	X	X	X
15–8	RESERVED		RO	0	X	X	X	X
7–0	VALUE	This value controls the number of RX Quick BD Chain entries that must be completed before a status block is generated. Setting this to 0 disables RX Event consumption from generating status blocks.	RW	0x1	X	X	X	X

HC RX TICKS REGISTER (HC_RX_TICKS, OFFSET 0x682c)*Table 350: HC RX Ticks Register (hc_rx_ticks, Offset 0x682c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–16	INT	This value controls the number of 1-μs ticks that will be counted for status block updates generated due to RX activity during interrupt processing. Setting this value to 0 disables the RX timer feature during interrupts.	RW	0	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X
9–0	VALUE	This value controls the number of 1-μs ticks that will be counted before a status block update is generated due to RX activity. This counter has two modes of operation. See the RX_TMR_MODE bit of the “ HC Configuration Register (hc_config, Offset 0x6808) ” on page 487. Setting this value to 0 disables the RX timer feature.	RW	0	X	X	X	X

HC TX TICKS REGISTER (HC_TX_TICKS, OFFSET 0x6830)*Table 351: HC TX Ticks Register (hc_tx_ticks, Offset 0x6830)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–16	INT	This value controls the number of 1-μs ticks that will be counted for status block updates generated due to TX activity during interrupt processing. Setting this value to 0 disables the TX timer feature during interrupts.	RW	0	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X
9–0	VALUE	This value controls the number of 1-μs ticks that will be counted before a status block update is generated due to TX activity. This counter has two modes of operation. See the TX_TMR_MODE bit of the “ HC Configuration Register (hc_config, Offset 0x6808) ” on page 487. Setting this value to 0 disables the TX timer feature.	RW	0	X	X	X	X

HC KERNEL COMPLETION TICKS REGISTER (HC_COM_TICKS, OFFSET 0x6834)*Table 352: HC Kernel Completion Ticks Register (hc_com_ticks, Offset 0x6834)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–16	INT	This value controls the number of 1-µs ticks that will be counted for status block updates generated due to Completion activity during interrupt processing. Setting this value to 0 disables the Completion timer feature during interrupts.	RW	0	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X
9–0	VALUE	This value controls the number of 1-µs ticks that will be counted before a status block update is generated due to Completion activity. This counter has two modes of operation. See the COM_TMR_MODE bit of the “ HC Configuration Register (hc_config, Offset 0x6808) ” on page 487. Setting this value to 0 disables the Completion timer feature during interrupts.	RW	0	X	X	X	X

HC COMMAND TICKS REGISTER (HC_CMD_TICKS, OFFSET 0x6838)*Table 353: HC Command Ticks Register (hc_cmd_ticks, Offset 0x6838)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–16	INT	This value controls the number of 1-µs ticks that will be counted for status block updates generated due to Command activity during interrupt processing. Setting this value to 0 disables the Command timer feature during interrupts.	RW	0	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X
9–0	VALUE	This value controls the number of 1-µs ticks that will be counted before a status block update is generated due to Command activity. This counter has two modes of operation. See the CMD_TMR_MODE bit of the “ HC Configuration Register (hc_config, Offset 0x6808) ” on page 487. Setting this value to 0 disables the Command timer feature during interrupts.	RW	0	X	X	X	X

HC PERIODIC TICKS REGISTER (HC_PERIODIC_TICKS, OFFSET 0x683c)*Table 354: HC Periodic Ticks Register (hc_periodic_ticks, Offset 0x683c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	–	–
31–16	HC_INT_PERIODIC_TICKS	This value controls the number of 1-μs ticks that will be counted between periodic checks for status block update need. This counter will count to the specified value, reload, and then check to see if any indexes are different than what was sent in the last status block. If they differ, then a status block will be generated. This counter counts during interrupts. This is used for heavy load situations and for when some periodic check is needed to back up the other algorithms. Setting this value to 0 disables the periodic timer feature.	RW	0	–	–	X	X
15–0	HC_PERIODIC_TICKS	This value controls the number of 1-μs ticks that will be counted between periodic checks for status block update need. This counter will count to the specified value, reload, and then check to see if any indexes are different than what was sent in the last status block. If they differ, then a status block will be generated. This counter counts the same both during and outside interrupts. This is used for heavy load situations and for when some periodic check is needed to back up the other algorithms. Setting this value to 0 disables the periodic timer feature.	RW	0	X	X	–	–
15–0	HC_PERIODIC_TICKS	This value controls the number of 1-μs ticks that will be counted between periodic checks for status block update need. This counter will count to the specified value, reload, and then check to see if any indexes are different than what was sent in the last status block. If they differ, then a status block will be generated. This counter counts outside interrupts. This is used for heavy load situations and for when some periodic check is needed to back up the other algorithms. Setting this value to 0 disables the periodic timer feature.	RW	0	–	–	X	X

HC STATISTICS COLLECT TICKS REGISTER (HC_STAT_COLLECT_TICKS, OFFSET 0x6840)*Table 355: HC Statistics Collect Ticks Register (hc_stat_collect_ticks, Offset 0x6840)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–12	RESERVED		RO	0	X	X	X	X
11–4	HC_STAT_COLL_TICKS	This register controls the number of 1-μs ticks that are counted between the start of each pass of collecting internal statistics from the register space. The resolution of this setting is limited to 16-μs intervals. Each time the counter expires, the internal statistics, enabled by the COLLECT_STATS bit (see “ HC Configuration Register (hc_config, Offset 0x6808) ” on page 487), is aggregated into the internal statistics memory. If this register is set to 0, then no statistics collection is done. The register range should provide up to a 4-ms period.	RW	0	X	X	X	X
3–0	RESERVED		RO	0	X	X	X	X

HC STATISTICS TICKS REGISTER (HC_STATS_TICKS, OFFSET 0x6844)*Table 356: HC Statistics Ticks Register (hc_stats_ticks, Offset 0x6844)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–8	HC_STAT_TICKS	This register controls the number of 1-μs ticks that are counted between the start of each pass of copying the internal statistics memory to host memory. Each time the counter expires, the statistics will be written to the address specified in the “ HC Statistics Address Low Register (hc_statistics_addr_l, Offset 0x6818) ” on page 490 and “ HC Statistics Address High Register (hc_statistics_addr_h, Offset 0x681c) ” on page 490. If this value is 0, the statistics DMA will be disabled. The register range should provide up to a 16 second period.	RW	0	X	X	X	X
7–0	RESERVED		RO	0	X	X	X	X

HC STATISTICS MEMORY DATA REGISTER (HC_STAT_MEM_DATA, OFFSET 0x684C)

Description	Mode	Reset	06	08	09	16
This register is used to access the data pointed to by the STAT_MEM_ADDR field of the RW “ HC Configuration Register (hc_config, Offset 0x6808) ” on page 487 in the statistics accumulation memory. Each time this register is read, the appropriate 32-bits of the memory will be read and the value returned. When this value is written, the value will be written to the STAT_MEM_ADDR field in the statistics accumulation memory.	RW	0	X	X	X	X

HC COALESCE NOW REGISTER (HC_COALESCE_NOW, OFFSET 0x6914)*Table 357: HC Coalesce Now Register (hc_coalesce_now, Offset 0x6914)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–30	RESERVED		RO	0	X	X	X	X
29–0	RESERVED		RO	0	X	X	–	–
29–21	COAL_ON_NXT_EVENT	This field is identical to the COAL_NOW field except that if there was an event after the last status block update a new status block update will be generated as quickly as possible. If there was no event after the last status block update the controller will wait for an event and then generate the status block. Each bit is associated with the corresponding status block (bit 21 for status block 0, bit 22 for status block 1, etc.).	SC	0	–	–	X	X
20	RESERVED		RO	0	–	–	X	X
19–11	COAL_NOW_WO_INT	This field is identical to the COAL_NOW field except that no interrupt will be generated after the status block has been generated. This may be used by a driver that wishes to poll some elements of the status block but continue to use interrupts for other elements. Each bit is associated with the corresponding status block (bit 19 for status block 0, bit 18 for status block 1, etc.).	SC	0	–	–	X	X
10	RESERVED		RO	0	–	–	X	X
9–1	COAL_NOW	Setting a bit in this field forces a status block to be generated as quickly as possible, regardless of any other HC settings. Each bit is associated with the corresponding status block (bit 9 for status block 0, bit 8 for status block 1, etc.).						
0	RESERVED		RO	0	–	–	X	X

HC MSI-X Bit Vector Register (HC_MSIX_BIT_VECTOR, OFFSET 0x6918)*Table 358: HC MSI-X Bit Vector Register (hc_msix_bit_vector, Offset 0x6918)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–9	RESERVED		RO	0	X	X	X	X
8–0	RESERVED		RO	0	X	X	–	–
8–0	VAL	This field is a bit mask that shows which MSI-X vectors are currently configured. The driver can override this by writing to this register.	RO	0	–	–	X	X

HC STATUS BLOCK CONFIGURATION 1 REGISTER (HC_SB_CONFIG_1, OFFSET 0x6A00)*Table 359: HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	—	—
23–20	PER_COLLECT_LIMIT	This field controls the number of status blocks for a particular processor that can be “stored” when the periodic collect feature is enabled. This limits the counter maximum that increments each time the periodic collect time expires when no status block was generated. Each count is a credit that can be used to generate more than one interrupt in a later periodic collect timer period.			—	—	X	X
19	RESERVED		RO	0	—	—	X	X
18	USE_INT_PARAM	When this bit is set the HC will automatically switch to using the interrupt mode parameters for a per-processor status block when an MSI/MSI-X is generated or in legacy interrupt mode when the sb_status register is read and the interrupt is masked for a particular status block.	RW	0	—	—	X	X
17	ONE_SHOT	When this bit is set one shot MSI/MSI-X mode is enabled which causes the HCTo automatically set the INT_MASK bit locally after an MSI or MSI-X is generated. This removes the need for the device driver to mask and unmask the interrupt for the next MSI or MSI-X message. This bit will also set the corresponding status block in the interrupt coalesce mode.	RW	0	—	—	X	X
16	PER_MODE	When this bit is set periodic collect mode is enabled which causes a status block to be generated after the collect timer expires following a periodic event.	RW	0	—	—	X	X
15–5	RESERVED		RO	0	—	—	X	X
4	CMD_TMR_MODE	This bit controls the Command Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	—	—	X	X

Table 359: HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
3	COM_TMR_MODE	This bit controls the Completion Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	-	-	X	X
2	TX_TMR_MODE	This bit controls the TX Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	-	-	X	X
1	RX_TMR_MODE	This bit controls the RX Timer index timer mode. If it is 0, then a status block will be generated at the specified time from the first event update since the last status block (collect mode). If it is 1, then a status block will be generated at the specified time from the last index update that has occurred (timeout mode).	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 1 REGISTER (HC_TX_QUICK_CONS_TRIP_1, OFFSET 0x6A04)**Table 360: HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	-	-
23–16	INT	The field controls the number of TX Quick BD chain entries that must be completed before a status block is generated during an interrupt.			-	-	X	X
15–8	RESERVED		RO	0	-	-	X	X
7–0	VALUE	This field controls the number of TX Quick BD chain entries that must be completed before a status block is generated. Clearing this field disables TX Quick BD chain consumption from generating status blocks.	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 1 REGISTER (HC_COMP_PROD_TRIP_1, OFFSET 0x6A08)*Table 361: HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	—	—
23–16	INT	The field controls the number of Kernel/Command Completion chain entries that must be completed before a status block is generated during an interrupt.	RW	0	—	—	X	X
15–8	RESERVED		RO	0	—	—	X	X
7–0	VALUE	This field controls the number of Kernel/Command Completion chain entries that must be completed before a status block is generated. Clearing this field disables Completion production from generating status blocks.	RW	0	—	—	X	X

HC RX QUICK CONSUMER TRIP 1 REGISTER (HC_RX_QUICK_CONS_TRIP_1, OFFSET 0x6A0C)*Table 362: HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	—	—
23–16	INT	The field controls the number of RX Quick BD chain entries that must be completed before a status block is generated during an interrupt.	RW	0	—	—	X	X
15–8	RESERVED		RO	0	—	—	X	X
7–0	VALUE	This field controls the number of RX Quick BD chain entries that must be completed before a status block is generated. Clearing this field disables RX Quick BD chain consumption from generating status blocks.	RW	0	—	—	X	X

HC RX TICKS 1 REGISTER (HC_RX_TICKS_1, OFFSET 0x6A10)*Table 363: HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–0	RESERVED		RO	0	X	X	—	—

Table 363: HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25–16	INT	The field controls the number of 1-µs ticks that will be counted before a status block is generated due to RX activity during interrupt processing. Clearing this value disables the RX timer feature during interrupts.	RW	0	—	—	X	X
15–10	RESERVED		RO	0	—	—	X	X
9–0	VALUE	The field controls the number of 1-µs ticks that will be counted for status block generated due to RX activity. This counter has two modes of operation. See the RX_TMR_MODE field of the “HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)” on page 497 . Clearing this value disables the RX timer feature.	RW	0	—	—	X	X

HC TX TICKS 1 REGISTER (HC_TX_TICKS_1, OFFSET 0x6A14)**Table 364: HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–0	RESERVED		RO	0	X	X	—	—
25–16	INT	The field controls the number of 1-µs ticks that will be counted before a status block is generated due to TX activity during interrupt processing. Clearing this value disables the TX timer feature during interrupts.	RW	0	—	—	X	X
15–10	RESERVED		RO	0	—	—	X	X
9–0	VALUE	The field controls the number of 1-µs ticks that will be counted for status block generated due to TX activity. This counter has two modes of operation. See the TX_TMR_MODE field of the “HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)” on page 497 . Clearing this value disables the TX timer feature.	RW	0	—	—	X	X

HC COMPLETION TICKS 1 REGISTER (HC_COM_TICKS_1, OFFSET 0x6A18)**Table 365: HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–0	RESERVED		RO	0	X	X	—	—

Table 365: HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25–16	INT	The field controls the number of 1-µs ticks that will be counted before a status block is generated due to Completion activity during interrupt processing. Clearing this value disables the Completion timer feature during interrupts.	RW	0	–	–	X	X
15–10	RESERVED		RO	0	–	–	X	X
9–0	VALUE	The field controls the number of 1-µs ticks that will be counted for status block generated due to Completion activity. This counter has two modes of operation. See the TMR_MODE field of the “HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)” on page 497. Clearing this value disables the Completion timer feature.	RW	0	–	–	X	X

HC COMMAND TICKS 1 REGISTER (HC_CMD_TICKS_1, OFFSET 0x6A1C)**Table 366: HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–0	RESERVED		RO	0	X	X	–	–
25–16	INT	The field controls the number of 1-µs ticks that will be counted before a status block is generated due to Command activity during interrupt processing. Clearing this value disables the Command timer feature during interrupts.	RW	0	–	–	X	X
15–10	RESERVED		RO	0	–	–	X	X
9–0	VALUE	The field controls the number of 1-µs ticks that will be counted for status block generated due to Command activity. This counter has two modes of operation. See the TMR_MODE field of the “HC Status Block Configuration 1 Register (hc_sb_config_1, Offset 0x6a00)” on page 497. Clearing this value disables the Command timer feature.	RW	0	–	–	X	X

HC PERIODIC TICKS 1 REGISTER (HC_PERIODIC_TICKS_1, OFFSET 0x6A20)**Table 367: HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X

Table 367: HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25–0	RESERVED		RO	0	X	X	–	–
25–16	INT	The field controls the number of 1-µs ticks that will be counted between periodic checks for status block updates. This counter will count to the specified value, reload, and then check to see if any indices have change since the last status block update. If any values have change then a new status block will be generated. This counter counts during interrupts. This is used for heavy load situations and when some periodic check is required to backup other algorithms. Clearing this value disables the periodic timer feature during interrupts.	RW	0	–	–	X	X
15–10	RESERVED		RO	0	–	–	X	X
9–0	VALUE	The field controls the number of 1-µs ticks that will be counted between periodic checks for status block updates. This counter will count to the specified value, reload, and then check to see if any indices have change since the last status block update. If any values have change then a new status block will be generated. This counter counts outside interrupts. This is used for heavy load situations and when some periodic check is required to backup other algorithms. Clearing this value disables the periodic timer feature during interrupts.	RW	0	–	–	X	X

HC STATUS BLOCK CONFIGURATION 2 REGISTER (HC_SB_CONFIG_2, OFFSET 0x6A24)**Table 368: HC Status Block Configuration 2 Register (hc_sb_config_2, Offset 0x6a24)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	–	–
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487 .	RW	0xf	–	–	X	X
19	RESERVED		RO	0	–	–	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487 .	RW	0	–	–	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487 .	RW	0	–	–	X	X



Table 368: HC Status Block Configuration 2 Register (hc_sb_config_2, Offset 0x6a24) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
15–5	RESERVED		RO	0	-	-	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	-	-	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 2 REGISTER (HC_TX_QUICK_CONS_TRIP_2, OFFSET 0x6A28)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 2 REGISTER (HC_COMP_PROD_TRIP_2, OFFSET 0x6A2C)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	-	-	X	X

HC RX QUICK CONSUMER TRIP 2 REGISTER (HC_RX_QUICK_CONS_TRIP_2, OFFSET 0x6A30)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX TICKS 2 REGISTER (HC_RX_TICKS_2, OFFSET 0x6A34)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX TICKS 2 REGISTER (HC_TX_TICKS_2, OFFSET 0x6A38)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 2 REGISTER (HC_COM_TICKS_2, OFFSET 0x6A3C)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 2 REGISTER (HC_CMD_TICKS_2, OFFSET 0x6A40)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 2 REGISTER (HC_PERIODIC_TICKS_2, OFFSET 0x6A44)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

HC STATUS BLOCK CONFIGURATION 3 REGISTER (HC_SB_CONFIG_3, OFFSET 0x6A48)*Table 369: HC Status Block Configuration 3 Register (hc_sb_config_3, Offset 0x6a48)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	-	-
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0xf	-	-	X	X
19	RESERVED		RO	0	-	-	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
15–5	RESERVED		RO	0	-	-	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X



Table 369: HC Status Block Configuration 3 Register (hc_sb_config_3, Offset 0x6a48) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	-	-	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 3 REGISTER (HC_TX_QUICK_CONS_TRIP_3, OFFSET 0x6A4C)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 3 REGISTER (HC_COMP_PROD_TRIP_3, OFFSET 0x6A50)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	-	-	X	X

HC RX QUICK CONSUMER TRIP 3 REGISTER (HC_RX_QUICK_CONS_TRIP_3, OFFSET 0x6A54)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX TICKS 3 REGISTER (HC_RX_TICKS_3, OFFSET 0x6A58)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX TICKS 3 REGISTER (HC_TX_TICKS_3, OFFSET 0x6A5C)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 3 REGISTER (HC_COM_TICKS_3, OFFSET 0x6A60)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 3 REGISTER (HC_CMD_TICKS_3, OFFSET 0x6A64)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 3 REGISTER (HC_PERIODIC_TICKS_3, OFFSET 0x6A68)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

HC STATUS BLOCK CONFIGURATION 4 REGISTER (HC_SB_CONFIG_4, OFFSET 0x6A6C)*Table 370: HC Status Block Configuration 4 Register (hc_sb_config_4, Offset 0x6a6c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	-	-
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0xf	-	-	X	X
19	RESERVED		RO	0	-	-	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
15–5	RESERVED		RO	0	-	-	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	-	-	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X



Table 370: HC Status Block Configuration 4 Register (hc_sb_config_4, Offset 0x6a6c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 4 REGISTER (HC_TX_QUICK_CONS_TRIP_4, OFFSET 0x6A70)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 4 REGISTER (HC_COMP_PROD_TRIP_4, OFFSET 0x6A74)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	-	-	X	X

HC RX QUICK CONSUMER TRIP 4 REGISTER (HC_RX_QUICK_CONS_TRIP_4, OFFSET 0x6A78)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX TICKS 4 REGISTER (HC_RX_TICKS_4, OFFSET 0x6A7C)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX TICKS 4 REGISTER (HC_TX_TICKS_4, OFFSET 0x6A80)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 4 REGISTER (HC_COM_TICKS_4, OFFSET 0x6A84)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 4 REGISTER (HC_CMD_TICKS_4, OFFSET 0x6A88)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 4 REGISTER (HC_PERIODIC_TICKS_4, OFFSET 0x6A8C)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

HC STATUS BLOCK CONFIGURATION 5 REGISTER (HC_SB_CONFIG_5, OFFSET 0x6A90)*Table 371: HC Status Block Configuration 5 Register (hc_sb_config_5, Offset 0x6a90)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	-	-
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0xf	-	-	X	X
19	RESERVED		RO	0	-	-	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
15–5	RESERVED		RO	0	-	-	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	-	-	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 5 REGISTER (HC_TX_QUICK_CONS_TRIP_5, OFFSET 0x6A94)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 5 REGISTER (HC_COMP_PROD_TRIP_5, OFFSET 0x6A98)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	-	-	X	X

HC RX QUICK CONSUMER TRIP 5 REGISTER (HC_RX_QUICK_CONS_TRIP_5, OFFSET 0x6A9C)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX TICKS 5 REGISTER (HC_RX_TICKS_5, OFFSET 0x6AA0)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX TICKS 5 REGISTER (HC_TX_TICKS_5, OFFSET 0x6AA4)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 5 REGISTER (HC_COM_TICKS_5, OFFSET 0x6AA8)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 5 REGISTER (HC_CMD_TICKS_5, OFFSET 0x6AAC)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 5 REGISTER (HC_PERIODIC_TICKS_5, OFFSET 0x6AB0)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

HC STATUS BLOCK CONFIGURATION 6 REGISTER (HC_SB_CONFIG_6, OFFSET 0x6AB4)*Table 372: HC Status Block Configuration 6 Register (hc_sb_config_6, Offset 0x6ab4)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	-	-
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0xf	-	-	X	X
19	RESERVED		RO	0	-	-	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
15–5	RESERVED		RO	0	-	-	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	-	-	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 6 REGISTER (HC_TX_QUICK_CONS_TRIP_6, OFFSET 0x6AB8)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 6 REGISTER (HC_COMP_PROD_TRIP_6, OFFSET 0x6ABC)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	-	-	X	X

HC RX QUICK CONSUMER TRIP 6 REGISTER (HC_RX_QUICK_CONS_TRIP_6, OFFSET 0x6AC0)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX TICKS 6 REGISTER (HC_RX_TICKS_6, OFFSET 0x6AC4)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX TICKS 6 REGISTER (HC_TX_TICKS_6, OFFSET 0x6AC8)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 6 REGISTER (HC_COM_TICKS_6, OFFSET 0x6ACC)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 6 REGISTER (HC_CMD_TICKS_6, OFFSET 0x6AD0)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 6 REGISTER (HC_PERIODIC_TICKS_6, OFFSET 0x6AD4)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

HC STATUS BLOCK CONFIGURATION 7 REGISTER (HC_SB_CONFIG_7, OFFSET 0x6AD8)*Table 373: HC Status Block Configuration 7 Register (hc_sb_config_7, Offset 0x6ad8)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	–	–
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0xf	–	–	X	X
19	RESERVED		RO	0	–	–	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	–	–	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	–	–	X	X
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	–	–	X	X
15–5	RESERVED		RO	0	–	–	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	–	–	X	X
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	–	–	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	–	–	X	X
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	–	–	X	X
0	RESERVED		RO	0	–	–	X	X

HC TX QUICK CONSUMER TRIP 7 REGISTER (HC_TX_QUICK_CONS_TRIP_7, OFFSET 0x6ADC)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	–	–	X	X

HC COMPLETION PRODUCER TRIP 7 REGISTER (HC_COMP_PROD_TRIP_7, OFFSET 0x6AE0)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	–	–	X	X

HC RX QUICK CONSUMER TRIP 7 REGISTER (HC_RX_QUICK_CONS_TRIP_7, OFFSET 0x6AE4)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX TICKS 7 REGISTER (HC_RX_TICKS_7, OFFSET 0x6AE8)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX TICKS 7 REGISTER (HC_TX_TICKS_7, OFFSET 0x6AEC)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 7 REGISTER (HC_COM_TICKS_7, OFFSET 0x6AF0)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 7 REGISTER (HC_CMD_TICKS_7, OFFSET 0x6AF4)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 7 REGISTER (HC_PERIODIC_TICKS_7, OFFSET 0x6AF8)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

HC STATUS BLOCK CONFIGURATION 8 REGISTER (HC_SB_CONFIG_8, OFFSET 0x6AFC)

Table 374: HC Status Block Configuration 7 Register (hc_sb_config_7, Offset 0x6ad8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–24	RESERVED		RO	0	X	X	X	X
23–0	RESERVED		RO	0	X	X	-	-
23–20	PER_COLLECT_LIMIT	See the PER_COLLECT_LIMIT field of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0xf	-	-	X	X



Table 374: HC Status Block Configuration 7 Register (hc_sb_config_7, Offset 0x6ad8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
19	RESERVED		RO	0	-	-	X	X
18	USE_INT_PARAM	See USE_INT_PARAM of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
17	ONE_SHOT	See ONE_SHOT of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
16	PER_MODE	See PER_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
15–5	RESERVED		RO	0	-	-	X	X
4	CMD_TMR_MODE	See CMD_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
3	COM_TMR_MODE	See COM_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487	RW	0	-	-	X	X
2	TX_TMR_MODE	See TX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
1	RX_TMR_MODE	See RX_TMR_MODE of "HC Configuration Register (hc_config, Offset 0x6808)" on page 487.	RW	0	-	-	X	X
0	RESERVED		RO	0	-	-	X	X

HC TX QUICK CONSUMER TRIP 8 REGISTER (HC_TX_QUICK_CONS_TRIP_8, OFFSET 0x6B00)

Description	Mode	Reset	06	08	09	16
See "HC TX Quick Consumer Trip 1 Register (hc_tx_quick_cons_trip_1, Offset 0x6a04)" on page 498	RW	0	-	-	X	X

HC COMPLETION PRODUCER TRIP 8 REGISTER (HC_COMP_PROD_TRIP_8, OFFSET 0x6B04)

Description	Mode	Reset	06	08	09	16
See "HC Completion Producer Trip 1 Register (hc_comp_prod_trip_1, Offset 0x6a08)" on page 499	RW	0	-	-	X	X

HC RX QUICK CONSUMER TRIP 8 REGISTER (HC_RX_QUICK_CONS_TRIP_8, OFFSET 0x6B08)

Description	Mode	Reset	06	08	09	16
See "HC RX Quick Consumer Trip 1 Register (hc_rx_quick_cons_trip_1, Offset 0x6a0c)" on page 499	RW	0	-	-	X	X

HC RX Ticks 8 REGISTER (HC_RX_TICKS_8, OFFSET 0x6B0C)

Description	Mode	Reset	06	08	09	16
See "HC RX Ticks 1 Register (hc_rx_ticks_1, Offset 0x6a10)" on page 499	RW	0	-	-	X	X

HC TX Ticks 8 REGISTER (HC_TX_TICKS_8, OFFSET 0x6B10)

Description	Mode	Reset	06	08	09	16
See "HC TX Ticks 1 Register (hc_tx_ticks_1, Offset 0x6a14)" on page 500	RW	0	-	-	X	X

HC COMPLETION TICKS 8 REGISTER (HC_COM_TICKS_8, OFFSET 0x6B14)

Description	Mode	Reset	06	08	09	16
See "HC Completion Ticks 1 Register (hc_com_ticks_1, Offset 0x6a18)" on page 500	RW	0	-	-	X	X

HC COMMAND TICKS 8 REGISTER (HC_CMD_TICKS_8, OFFSET 0x6B18)

Description	Mode	Reset	06	08	09	16
See "HC Command Ticks 1 Register (hc_cmd_ticks_1, Offset 0x6a1c)" on page 501	RW	0	-	-	X	X

HC PERIODIC TICKS 8 REGISTER (HC_PERIODIC_TICKS_8, OFFSET 0x6B1C)

Description	Mode	Reset	06	08	09	16
See "HC Periodic Ticks 1 Register (hc_periodic_ticks_1, Offset 0x6a20)" on page 501	RW	0	-	-	X	X

TRANSMIT PROCESSOR (TXP) REGISTERS

The job of the TX Processor (TXP) is to segment the slot request into packets with compliance to the protocols. All the transmit protocol processing, including header generation, on the offloaded TCP connections are performed here. The TX Processor also decides the exact number of bytes to be transmitted. The TX Processor provides maximum flexibility to adapt to future changes in the protocol. The TX Processor does not have access to the actual data to be transmitted, which are DMAed into the chip by the TX DMA located downstream. This arrangement forms an efficient transmit pipeline which allows execution of lengthy protocol processing and the data movement across PCI bus to be overlapped without wasting any PCI bus bandwidth or internal buffering for transmit.

TXP CPU MODE REGISTER (TXP_CPU_MODE, OFFSET 0x45000)

Table 375: TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	SPAD_UNDERFLOW_HALT_ENA	When this bit is set, the CPU will halt when the SPAD_UNDERFLOW bit of the “TXP CPU State Register (txp_cpu_state, Offset 0x45004)” on page 834 is set.	RW	0x1	X	X	X	X
14	RESERVED		RO	0	X	X	X	X
13	FIO_ABORT_HALT_ENA	When this bit is set, the CPU will halt when an abort is indicated from any Fast IO space peripheral.	RW	0x1	X	X	X	X
12	BAD_INST_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_PC_HALTED field of the “TXP CPU State Register (txp_cpu_state, Offset 0x45004)” on page 834 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
11	BAD_DATA_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_DATA_ADDR_HALTED bit in the “TXP CPU State Register (txp_cpu_state, Offset 0x45004)” on page 834 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
10	SOFT_HALT	When this bit is set, the CPU will halt. This bit is cleared by an exception or reset. If the processor does not have a ROM, then this bit will reset to 1 so that no code is executed from the scratchpad. If the processor does have a ROM, this bit resets to 0 so that the processor executes from ROM after reset.	RW	0	X	X	X	X
9–8	RESERVED		RO	0	X	X	X	X

Table 375: TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
7	INTERRUPT_ENA	When this bit is set to 1, the interrupt is enabled. When this bit is 0, any interrupt will be ignored. This bit can also be set by writing the “ TXP CPU Interrupt Enable Register (txp_cpu_interrupt_enable, Offset 0x45028) ” on page 837.	RW	0	X	X	X	X
6	MSG_BIT1	This is a simple RW bit.	RW	0x1	X	X	X	X
5-4	RESERVED		RO	0	X	X	X	X
3	PAGE_0_INST_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_INST_HALTED bit of the “ TXP CPU State Register (txp_cpu_state, Offset 0x45004) ” on page 834 when an instruction references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
2	PAGE_0_DATA_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_DATA_HALTED bit of the “ TXP CPU State Register (txp_cpu_state, Offset 0x45004) ” on page 834 when data references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
1	STEP_ENA	When this bit is set, the processor is allowed to execute one cycle regardless of any halt conditions. If the halting condition still exists, the CPU will halt again after the one cycle, otherwise, it will resume normal operation.	RW	0	X	X	X	X
0	LOCAL_RST	When this bit is written to a 1, the processor will reset as if from power-up state. All Reset values of registers will be assigned.	RW	0	X	X	X	X

TXP CPU STATE REGISTER (TXP_CPU_STATE, OFFSET 0x45004)**Table 376: TXP CPU State Register (txp_cpu_state, Offset 0x45004)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BLOCKED_READ	This bit indicates that a blocking data cache miss occurred, causing the CPU to stall while data is fetched from memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred. Writing a 1 clears this bit.	WC	0	X	X	X	X
30-16	RESERVED		RO	0	X	X	X	X

Table 376: TXP CPU State Register (txp_cpu_state, Offset 0x45004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
15	INST_FETCH_STALL	This bit is set while the processor is stalled due to instruction fetch.	RO	0	X	X	X	X
14	DATA_ACCESS_STALL	This bit is set while the processor is stalled due to data access.	RO	0	X	X	X	X
13	RESERVED		RO	0	X	X	X	X
12	INTERRRUPT	This bit is set each time an interrupt input is asserted, regardless of the interrupt enable bit (see the INTERRUPT_ENA field of the "TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000)" on page 516).	WC	0	X	X	X	X
11	SPAD_UNDERFLOW	This bit is set each time an attempt is made to access the underflow area of the Scratchpad.	WC	0	X	X	x	X
10	SOFT_HALTED	This bit is set while the processor is halted due to the setting of the SOFT_HALT bit of the "TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000)" on page 516 .	RO	0	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_HALTED	This bit is set while the processor is halted due to the generation of an abort condition by one or more Fast IO devices within the CPU block. This will only happen if halt is enabled by the FIO_ABORT_HALT_ENA bit of the "TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000)" on page 516 .	RO	0	X	X	X	X
7	ALIGN_HALTED	This bit is set while the processor is halted due to bad memory alignment problem on a load or store instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
6	BAD_PC_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED	This bit is set while the processor is halted due to bad data reference address. Writing a 1 clears this bit.	WC	0	X	X	X	X
4	PAGE_0_INST_HALTED	This bit is set while the processor is halted due to executing an instruction within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
3	PAGE_0_DATA_HALTED	This bit is set while the processor is halted due to accessing data within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
2	BAD_INST_HALTED	This bit is set while the processor is halted due fetching an invalid instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	x

Table 376: TXP CPU State Register (txp_cpu_state, Offset 0x45004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
0	BREAKPOINT	This bit is set while the processor is halted due reaching a hardware breakpoint as enabled in the mode register. Writing a 1 clears this bit.	WC	0	X	X	X	X

TXP CPU EVENT MASK (TXP_CPU_EVENT_MASK, OFFSET 0x45008)

This register provides a corresponding bit for each field in the “[TXP CPU State Register \(txp_cpu_state, Offset 0x45004\)](#)” on page 517 to enable it into the equation for generating the TX Processor Attention output. The reset value of 1 masks all halt conditions from generating an attention.

Table 377: TXP CPU Event Mask (txp_cpu_event_mask, Offset 0x45008)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–13	RESERVED	—	RO	0	X	X	X	X
12	INTERRUPT_MASK	—	RW	0	X	X	X	X
11	SPAD_UNDERFLOW_MASK	—	RW	0	X	X	X	X
10	SOFT_HALTED_MASK	—	RW	0x1	X	X	X	X
9	RESERVED	—	RO	0	X	X	X	X
8	FIO_ABORT_MASK	—	RW	0x1	X	X	X	X
7	ALIGN_HALTED_MASK	—	RW	0	X	X	X	X
6	BAD_PC_HALTED_MASK	—	RW	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED_MASK	—	RW	0	X	X	X	X
4	PAGE_0_INST_HALTED_MASK	—	RW	0	X	X	X	X
3	PAGE_0_DATA_HALTED_MASK	—	RW	0	X	X	X	X
2	BAD_INST_HALTED_MASK	—	RW	0	X	X	X	X
1	RESERVED	—	RO	0	X	X	X	X
0	BREAKPOINT_MASK	—	RW	0	X	X	X	X

TXP CPU PROGRAM COUNTER REGISTER (TXP_CPU_PROGRAM_COUNTER, OFFSET 0x4501c)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

TXP CPU REGISTER FILE REGISTERS (TXP_CPU_REG_FILE[32], OFFSET 0x45200)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

TXP CPU FTQ DATA REGISTERS (TXP_FTQ_DATA[14], OFFSET 0x453c0)

Description	Mode	Reset	06	08	09	16
	RO	0	X	X	X	X

TXP CPU FTQ COMMAND REGISTER (TXP_FTQ_CMD, OFFSET 0x453f8)

The TX Processor FTQ is 16 records deep.

Table 378: TXP CPU FTQ Command Register (txp_ftq_cmd, Offset 0x453f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “TXP CPU Mode Register (txp_cpu_mode, Offset 0x45000)” on page 516. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the “TXP CPU Data Access Register (txp_cpu_data_access, Offset 0x45024)” on page 837 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the “TXP CPU Data Access Register (txp_cpu_data_access, Offset 0x45024)” on page 837 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “TXP CPU Data Access Register (txp_cpu_data_access, Offset 0x45024)” on page 837 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X



Table 378: TXP CPU FTQ Command Register (txp_ftq_cmd, Offset 0x453f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16									
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X									
24–11	RESERVED		RO	0	X	X	X	X									
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “TXP CPU Data Access Register (txp_cpu_data_access, Offset 0x45024)” on page 837.	RW	0	X	X	X	X									
<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Value</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </tbody> </table>									Value	Name	Description	0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
Value	Name	Description															
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.															
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.															
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “TXP CPU Data Access Register (txp_cpu_data_access, Offset 0x45024)” on page 837.	RW	0	X	X	X	X									

TXP CPU FTQ CONTROL REGISTER (TXP_FTQ_CTL, OFFSET 0x453FC)

Table 379: TXP CPU FTQ Control Register (txp_ftq_ctl, Offset 0x453fc)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X

Table 379: TXP CPU FTQ Control Register (txp_ftq_ctl, Offset 0x453fc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

TXP CPU SCRATCHPAD AREA (TXP_SCRATCH[8192], OFFSET 0x60000)

Description	Mode	Reset	06	08	09	16
This register space is the TX Processor scratch pad space that is visible at 0x0 by RW the processor. This can be modified at any time and may be used for processor-to-processor communication.	0		X	X	X	X

TRANSMIT PATCH-UP (TPAT) REGISTERS

The purpose of the TX Patch-Up (TPAT) processor is to perform any TX operations that must be done after the data has been moved into the chip. These jobs include the function of updating (“patching”) the holes in the TX Payload Queue and updating the checksum in the TX Header Queue.

TPAT CPU MODE REGISTER (TPAT_CPU_MODE, OFFSET 0x85000)

Table 380: TPAT CPU Mode Register (tpat_cpu_mode, Offset 0x85000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	SPAD_UNDERFLOW_HALT_ENA	When this bit is set, the CPU will halt when the SPAD_UNDERFLOW bit of the “TPAT CPU State Register (tpat_cpu_state, Offset 0x85004)” on page 844 is set.	RW	0x1	X	X	X	X
14	RESERVED		RO	0	X	X	X	X
13	FIO_ABORT_HALT_ENA	When this bit is set, the CPU will halt when an abort is indicated from any Fast IO space peripheral.	RW	0x1	X	X	X	X
12	BAD_INST_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_PC_HALTED field of the “TPAT CPU State Register (tpat_cpu_state, Offset 0x85004)” on page 844 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
11	BAD_DATA_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_DATA_ADDR_HALTED bit in the “TPAT CPU State Register (tpat_cpu_state, Offset 0x85004)” on page 844 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
10	SOFT_HALT	When this bit is set, the CPU will halt. This bit is cleared by an exception or reset. If the processor does not have a ROM, then this bit will reset to set so that no code is executed from the scratchpad. If the processor does have a ROM, this bit resets a cleared so that the processor executes from ROM after reset.	RW	0	X	X	X	X
9–8	RESERVED		RO	0	X	X	X	X
7	INTERRUPT_ENA	When this bit is set to 1, the interrupt is enabled. When this bit is 0, any interrupt will be ignored. This bit can also be set by writing the “TPAT CPU Interrupt Enable Register (tpat_cpu_interrupt_enable, Offset 0x85028)” on page 847	RW	0	X	X	X	X

Table 380: TPAT CPU Mode Register (tpat_cpu_mode, Offset 0x85000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
6	MSG_BIT1	This is a simple RW bit.	RW	0x1	X	X	X	X
5–4	RESERVED		RO	0	X	X	X	X
3	PAGE_0_INST_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_INST_HALTED bit of the "TPAT CPU State Register (tpat_cpu_state, Offset 0x85004)" on page 844 when an instruction references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
2	PAGE_0_DATA_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_DATA_HALTED bit of the "TPAT CPU State Register (tpat_cpu_state, Offset 0x85004)" on page 844 when data references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
1	STEP_ENA	When this bit is set, the processor is allowed to execute one cycle regardless of any halt conditions. If the halting condition still exists, the CPU will halt again after the one cycle, otherwise, it will resume normal operation.	RW	0	X	X	X	X
0	LOCAL_RST	When this bit is written to a 1, the processor will reset as if from power-up state. All Reset value of registers will be assigned.	RW	0	X	X	X	X

TPAT CPU STATE REGISTER (TPAT_CPU_STATE, OFFSET 0x85004)**Table 381: TPAT CPU State Register (tpat_cpu_state, Offset 0x85004)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BLOCKED_READ	This bit indicates that a blocking data cache miss occurred, causing the CPU to stall while data is fetched from memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred. Writing a 1 clears this bit.	WC	0	X	X	X	X
30–16	RESERVED		RO	0	X	X	X	X
15	INST_FETCH_STALL	This bit is set while the processor is stalled due to instruction fetch.	RO	0	X	X	X	X
14	DATA_ACCESS_STALL	This bit is set while the processor is stalled due to data access.	RO	0	X	X	X	X
13	RESERVED		RO	0	X	X	X	X

Table 381: TPAT CPU State Register (*tpat_cpu_state*, Offset 0x85004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	INTERRRUPT	This bit is set each time an interrupt input is asserted, regardless of the interrupt enable bit (see the INTERRUPT_ENA field of the “TPAT CPU Mode Register (<i>tpat_cpu_mode</i> , Offset 0x85000)” on page 523).	WC	0	X	X	X	X
11	SPAD_UNDERFLOW	This bit is each time an attempt is made to access the underflow area of the Scratchpad.	WC	0	X	X	X	X
10	SOFT_HALTED	This bit is set while the processor is halted due to the setting of the SOFT_HALT bit of the “TPAT CPU Mode Register (<i>tpat_cpu_mode</i> , Offset 0x85000)” on page 523.	RO	0	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_HALTED	This bit is set while the processor is halted due to the generation of an abort condition by one or more Fast IO devices within the CPU block. This will only happen if halt is enabled by the FIO_ABORT_HALT_ENA bit of the “TPAT CPU Mode Register (<i>tpat_cpu_mode</i> , Offset 0x85000)” on page 523.	RO	0	X	X	X	X
7	ALIGN_HALTED	This bit is set while the processor is halted due to bad memory alignment problem on a load or store instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
6	BAD_PC_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED	This bit is set while the processor is halted due to bad data reference address. Writing a 1 clears this bit.	WC	0	X	X	X	X
4	PAGE_0_INST_HALTED	This bit is set while the processor is halted due to executing an instruction within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
3	PAGE_0_DATA_HALTED	This bit is set while the processor is halted due to accessing data within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
2	BAD_INST_HALTED	This bit is set while the processor is halted due fetching an invalid instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT	This bit is set while the processor is halted due reaching a hardware breakpoint as enabled in the mode register. Writing a 1 clears this bit.	WC	0	X	X	X	X

TPAT CPU EVENT MASK REGISTER (TPAT_CPU_EVENT_MASK, OFFSET 0x85008)

This register provides one bit for each field in the “[TPAT CPU State Register \(tpat_cpu_state, Offset 0x85004\)](#)” on page 524 to enable it into the equation for generation the TX Patchup Processor Attention output. The reset value of 1 masks all halt conditions from generating an attention.

Table 382: TPAT CPU Event Mask Register (tpat_cpu_event_mask, Offset 0x85008)

Bit	Name	Description	Mode	Reset	06	08	09	16
31-13	RESERVED		RO	0	X	X	X	X
12	INTERRUPT_MASK	—	RW	0	X	X	X	X
11	SPAD_UNDERFLOW_MASK	—	RW	0	X	X	X	X
10	SOFT_HALTED_MASK	—	RW	0x1	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_MASK	—	RW	0x1	X	X	X	X
7	ALIGN_HALTED_MASK	—	RW	0	X	X	X	X
6	BAD_PC_HALTED_MASK	—	RW	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED_MASK	—	RW	0	X	X	X	X
4	PAGE_0_INST_HALTED_MASK	—	RW	0	X	X	X	X
3	PAGE_0_DATA_HALTED_MASK	—	RW	0	X	X	X	X
2	BAD_INST_HALTED_MASK	—	RW	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT_MASK	—	RW	0	X	X	X	X

TPAT CPU PROGRAM COUNTER REGISTER (TPAT_CPU_PROGRAM_COUNTER, OFFSET 0x8501c)

Description	Mode	Reset	06	08	09	16
This register allows the program counter to read at any time. The value can be modified RW when the processor is halted. Writes will also clear any pending instruction in the decode stage of the pipeline. Bits 31-2 are implemented. 1s written to bits 1-0 are ignored. If the processor has a ROM, then the reset value of this register points to the internal ROM. If the processor does not have a ROM, then this reset value points to the scratchpad area.	0		X	X	X	X

TPAT CPU REGISTER FILE REGISTERS (TPAT_CPU_REG_FILE[32], OFFSET 0x85200)

Description	Mode	Reset	06	08	09	16
While the processor is halted, the general-purpose processor registers (r0-r31) can be RW read and written through these register locations.	0		X	X	X	X

TPAT FTQ DATA REGISTERS (TPAT_FTQ_DATA[14], OFFSET 0x853c0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	0	X X X X				

TPAT FTQ COMMAND REGISTER (TPAT_FTQ_CMD, OFFSET 0x853f8)

The TX Patchup Processor FTQ is 16 records deep.

Table 383: TPAT FTQ Command Register (tpat_ftq_cmd, Offset 0x853f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X X X X			
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “ TPAT CPU Mode Register (tpat_cpu_mode, Offset 0x85000) ” on page 523. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X X X X			
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “ TPAT CPU Mode Register (tpat_cpu_mode, Offset 0x85000) ” on page 523. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X X X X			
28	ADD_DATA	When this bit is written as a 1, the “ TPAT FTQ Data Registers (tpat_ftq_data[14], Offset 0x853c0) ” on page 527 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X X X X			
27	ADD_INTERVEN	When this bit is written as a 1, the TPAT FTQ Data Registers (tpat_ftq_data[14], Offset 0x853c0) is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X X X X			

Table 383: TPAT FTQ Command Register (tpat_ftq_cmd, Offset 0x853f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16						
26	RD_DATA	When this bit is written as a 1, the “TPAT FTQ Data Registers (tpat_ftq_data[14], Offset 0x853c0)” on page 527 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X						
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X						
24–11	RESERVED		RO	0	X	X	X	X						
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “TPAT FTQ Data Registers (tpat_ftq_data[14], Offset 0x853c0)” on page 527.	RW	0	X	X	X	X						
Value Name Description														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">0</td> <td style="width: 10%;">0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </table>									0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.												
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.												
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the TPAT FTQ Data Registers (tpat_ftq_data[14], Offset 0x853c0).	RW	0	X	X	X	X						

TPAT FTQ CONTROL REGISTER (TPAT_FTQ_CTL, OFFSET 0x853FC)*Table 384: TPAT FTQ Control Register (tpat_ftq_ctl, Offset 0x853fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

TPAT SCRATCHPAD AREA (TPAT_SCRATCH[4096], OFFSET 0xA0000)

Description	Mode	Reset	06	08	09	16
The TPAT scratchpad area is visible at 0x0 by the TPAT processor. This area can be RO modified at any time and may be used for processor-to-processor communication	0	X	—	—	—	—

TPAT SCRATCHPAD AREA (TPAT_SCRATCH[3072], OFFSET 0xA0000)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
The TPAT scratchpad area is visible at 0x0 by the TPAT processor. This area can be RO modified at any time and may be used for processor-to-processor communication.	0	-	X	X	X	

RECEIVE PROCESSOR (RXP) REGISTERS

RXP CPU MODE REGISTER (RXP_CPU_MODE, OFFSET 0xC5000)

Table 385: RXP CPU Mode Register (rxp_cpu_mode, Offset 0xc5000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	SPAD_UNDERFLOW_HALT_ENA	When this bit is set, the CPU will halt when the SPAD_UNDERFLOW bit of the is set.	RW	0x1	X	X	X	X
14	RESERVED		RO	0	X	X	X	X
13	FIO_ABORT_HALT_ENA	When this bit is set, the CPU will halt when an abort is indicated from any Fast IO space peripheral.	RW	0x1	X	X	X	X
12	BAD_INST_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_PC_HALTED field of the “ RXP CPU State Register (rxp_cpu_state, Offset 0xc5004) ” on page 854 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
11	BAD_DATA_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_DATA_ADDR_HALTED bit in the “ RXP CPU State Register (rxp_cpu_state, Offset 0xc5004) ” on page 854 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
10	SOFT_HALT	When this bit is set, the CPU will halt. This bit is cleared by an exception or reset. If the processor does not have a ROM, then this bit will reset to set so that no code is executed from the scratchpad. If the processor does have a ROM, this bit resets a cleared so that the processor executes from ROM after reset.	RW	0	X	X	X	X
9–8	RESERVED		RO	0	X	X	X	X
7	INTERRUPT_ENA	When this bit is set to 1, the interrupt is enabled. When this bit is 0, any interrupt will be ignored. This bit can also be set by writing the “ RXP CPU Interrupt Enable Register (rxp_cpu_interrupt_enable, Offset 0xc5028) ” on page 857	RW	0	X	X	X	X
6	MSG_BIT1	This is a simple RW bit.	RW	0x1	X	X	X	X
5–4	RESERVED		RO	0	X	X	X	X

Table 385: RXP CPU Mode Register (rxp_cpu_mode, Offset 0xc5000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
3	PAGE_0_INST_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_INST_HALTED bit of the "RXP CPU State Register (rpx_cpu_state, Offset 0xc5004)" on page 854 when an instruction references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
2	PAGE_0_DATA_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_DATA_HALTED bit of the "RXP CPU State Register (rpx_cpu_state, Offset 0xc5004)" on page 854 when data references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
1	STEP_ENA	When this bit is set, the processor is allowed to execute one cycle regardless of any halt conditions. If the halting condition still exists, the CPU will halt again after the one cycle, otherwise, it will resume normal operation.	RW	0	X	X	X	X
0	LOCAL_RST	When this bit is written to a 1, the processor will reset as if from power-up state. All Reset value of registers will be assigned.	RW	0	X	X	X	X

RXP CPU STATE REGISTER (RXP_CPU_STATE, OFFSET 0XC5004)**Table 386: RXP CPU State Register (rpx_cpu_state, Offset 0xc5004)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BLOCKED_READ	This bit indicates that a blocking data cache miss occurred, causing the CPU to stall while data is fetched from memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred. Writing a 1 clears this bit.	WC	0	X	X	X	X
30–16	RESERVED		RO	0	X	X	X	X
15	INST_FETCH_STALL	This bit is set while the processor is stalled due to instruction fetch.	RO	0	X	X	X	X
14	DATA_ACCESS_STALL	This bit is set while the processor is stalled due to data access.	RO	0	X	X	X	X
13	RESERVED		RO	0	X	X	X	X

Table 386: RXP CPU State Register (*rxp_cpu_state*, Offset 0xc5004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	INTERRRUPT	This bit is set each time an interrupt input is asserted, regardless of the interrupt enable bit (see the INTERRUPT_ENA field of the “RXP CPU Mode Register (<i>rxp_cpu_mode</i> , Offset 0xc5000)” on page 531).	WC	0	X	X	X	X
11	SPAD_UNDERFLOW	This bit is each time an attempt is made to access the underflow area of the Scratchpad.	WC	0	X	X	X	X
10	SOFT_HALTED	This bit is set while the processor is halted due to the setting of the SOFT_HALT bit of the “RXP CPU Mode Register (<i>rxp_cpu_mode</i> , Offset 0xc5000)” on page 531.	RO	0	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_HALTED	This bit is set while the processor is halted due to the generation of an abort condition by one or more Fast IO devices within the CPU block. This will only happen if halt is enabled by the FIO_ABORT_HALT_ENA bit of the “RXP CPU Mode Register (<i>rxp_cpu_mode</i> , Offset 0xc5000)” on page 531.	RO	0	X	X	X	X
7	ALIGN_HALTED	This bit is set while the processor is halted due to bad memory alignment problem on a load or store instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
6	BAD_PC_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
4	PAGE_0_INST_HALTED	This bit is set while the processor is halted due to executing an instruction within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
3	PAGE_0_DATA_HALTED	This bit is set while the processor is halted due to accessing data within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
2	BAD_INST_HALTED	This bit is set while the processor is halted due fetching an invalid instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT	This bit is set while the processor is halted due reaching a hardware breakpoint as enabled in the mode register. Writing a 1 clears this bit.	WC	0	X	X	X	X

RXP CPU EVENT MASK REGISTER (RXP_CPU_EVENT_MASK, OFFSET 0xc5008)

This register provides one bit for each “[RXP CPU State Register \(rxp_cpu_state, Offset 0xc5004\)](#)” on page 532 bit to enable it into the equation for generation the RX Processor Attention output. The reset value of 1 masks all halt conditions from generating an attention.

Table 387: RXP CPU Event Mask Register (rxp_cpu_event_mask, Offset 0xc5008)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–13	RESERVED		RO	0	X	X	X	X
12	INTERRUPT_MASK	—	RW	0	X	X	X	X
11	SPAD_UNDERFLOW_MASK	—	RW	0	X	X	X	X
10	SOFT_HALTED_MASK	—	RW	0x1	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_MASK	—	RW	0x1	X	X	X	X
7	ALIGN_HALTED_MASK	—	RW	0	X	X	X	X
6	BAD_PC_HALTED_MASK	—	RW	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED_MASK	—	RW	0	X	X	X	X
4	PAGE_0_INST_HALTED_MASK	—	RW	0	X	X	X	X
3	PAGE_0_DATA_HALTED_MASK	—	RW	0	X	X	X	X
2	BAD_INST_HALTED_MASK	—	RW	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT_MASK	—	RW	0	X	X	X	X

RXP CPU PROGRAM COUNTER REGISTER (RXP_CPU_PROGRAM_COUNTER, OFFSET 0xc501c)

Description	Mode	Reset	06	08	09	16
This register allows the program counter to read at any time. The value can be modified RW when the processor is halted. Writes will also clear any pending instruction in the decode stage of the pipeline. Bits 31–2 are implemented. 1s written to bits 1–0 are ignored. If the processor has a ROM, then the reset value of this register points to the internal ROM. If the processor does not have a ROM, then this reset value points to the scratchpad area.	0		X	X	X	X

RXP CPU REGISTER FILE REGISTERS (RXP_CPU_REG_FILE[32], OFFSET 0xc5200)

Description	Mode	Reset	06	08	09	16
While the processor is halted, the general-purpose processor registers (r0–r31) can be RW read and written through these register locations.	0		X	X	X	X

RXP CFTQ DATA REGISTERS (RXP_CFTQ_DATA[14], OFFSET 0xC5380)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	0	X X X X				

RXP CFTQ COMMAND REGISTER (RXP_CFTQ_CMD, OFFSET 0xC53B8)

The RX Processor Command FTQ is 32 records deep.

Table 388: RXP CFTQ Command Register (rpx_cftq_cmd, Offset 0xc53b8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X X X X			
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “ RXP CFTQ Control Register (rpx_cftq_ctl, Offset 0xc53bc) ” on page 537. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X X X X			
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “ RXP CFTQ Control Register (rpx_cftq_ctl, Offset 0xc53bc) ” on page 537. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X X X X			
28	ADD_DATA	When this bit is written as a 1, the “ RXP CFTQ Data Registers (rpx_cftq_data[14], Offset 0xc5380) ” on page 535 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X X X X			
27	ADD_INTERVEN	When this bit is written as a 1, the “ RXP CFTQ Data Registers (rpx_cftq_data[14], Offset 0xc5380) ” on page 535 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X X X X			

Table 388: RXP CFTQ Command Register (rxp_cftq_cmd, Offset 0xc53b8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16						
26	RD_DATA	When this bit is written as a 1, the “RXP CFTQ Data Registers (rxp_cftq_data[14], Offset 0xc5380)” on page 535 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X						
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X						
24–11	RESERVED		RO	0	X	X	X	X						
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “RXP CFTQ Data Registers (rxp_cftq_data[14], Offset 0xc5380)” on page 535.	RW	0	X	X	X	X						
Val Name Description														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">0</td> <td style="width: 10%;">0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </table>									0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.												
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.												
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “RXP CFTQ Control Register (rxp_cftq_ctl, Offset 0xc53bc)” on page 537.	RW	0	X	X	X	X						

RXP CFTQ CONTROL REGISTER (RXP_CFTQ_CTL, OFFSET 0xC53BC)*Table 389: RXP CFTQ Control Register (rxp_cftq_ctl, Offset 0xc53bc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

RXP FTQ DATA REGISTERS (RXP_FTQ_DATA[14], OFFSET 0xC53C0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this block.	RO	0	X	X	X	X

RXP FTQ COMMAND REGISTER (RXP_FTQ_CMD, OFFSET 0xC53F8)

The RX Processor Main FTQ is 256 records deep.

Table 390: RXP FTQ Command Register (rxp_ftq_cmd, Offset 0xc53f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)" on page 539 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)" on page 539 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the "RXP FTQ Data Registers (rxp_ftq_data[14], Offset 0xc53c0)" on page 537 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the "RXP FTQ Data Registers (rxp_ftq_data[14], Offset 0xc53c0)" on page 537 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the "RXP FTQ Data Registers (rxp_ftq_data[14], Offset 0xc53c0)" on page 537 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 390: RXP FTQ Command Register (*rxp_ftq_cmd*, Offset 0xc53f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16								
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X								
24–11	RESERVED		RO	0	X	X	X	X								
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “RXP FTQ Data Registers (<i>rxp_ftq_data[14]</i> , Offset 0xc53c0)” on page 537.	RW	0	X	X	X	X								
		<table border="1"> <thead> <tr> <th>Val</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </tbody> </table>	Val	Name	Description	0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.					
Val	Name	Description														
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.														
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.														
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “RXP FTQ Control Register (<i>rxp_ftq_ctl</i> , Offset 0xc53fc)” on page 539.	RW	0	X	X	X	X								

RXP FTQ CONTROL REGISTER (RXP_FTQ_CTL, OFFSET 0xC53FC)Table 391: RXP FTQ Control Register (*rxp_ftq_ctl*, Offset 0xc53fc)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X

Table 391: RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
21-12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11-3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

RXP SCRATCHPAD AREA (RXP_SCRATCH[8192], OFFSET 0xE0000)

Description	Mode	Reset	06	08	09	16
This is the RXP scratchpad area which is visible at 0x0 by the RXP processor. RW This read/write area can be modified at any time and may be used for processor-to-processor communication.	0	X	-	-	-	-

RXP SCRATCHPAD AREA (RXP_SCRATCH[9216], OFFSET 0xE0000)

Description	Mode	Reset	06	08	09	16
This is the RXP scratchpad area which is visible at 0x0 by the RXP processor. RW This read/write area can be modified at any time and may be used for processor-to-processor communication.	0	- X - -				

RXP SCRATCHPAD AREA (RXP_SCRATCH[10240], OFFSET 0xE0000)

Description	Mode	Reset	06	08	09	16
This is the RXP scratchpad area which is visible at 0x0 by the RXP processor. RW This read/write area can be modified at any time and may be used for processor-to-processor communication.	0	- - X X				

COMPLETION PROCESSOR (CP) REGISTERS

COM CPU MODE REGISTER (COM_CPU_MODE, OFFSET 0x105000)

Table 392: COM CPU Mode Register (com_cpu_mode, Offset 0x105000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	SPAD_UNDERFLOW_HALT_ENA	When this bit is set, the CPU will halt when the SPAD_UNDERFLOW bit of the “RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)” on page 539 is set.	RW	0x1	X	X	X	X
14	RESERVED		RO	0	X	X	X	X
13	FIO_ABORT_HALT_ENA	When this bit is set, the CPU will halt when an abort is indicated from any Fast I/O space peripheral.	RW	0x1	X	X	X	X
12	BAD_INST_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_PC_HALTED field of the “RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)” on page 539 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
11	BAD_DATA_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_DATA_ADDR_HALTED bit in the “RXP FTQ Control Register (rxp_ftq_ctl, Offset 0xc53fc)” on page 539 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
10	SOFT_HALT	When this bit is set, the CPU will halt. This bit is cleared by an exception or reset. If the processor does not have a ROM, then this bit will reset to set so that no code is executed from the scratchpad. If the processor does have a ROM, this bit resets a cleared so that the processor executes from ROM after reset.	RW	0	X	X	X	X
9–8	RESERVED		RO	0	X	X	X	X
7	INTERRUPT_ENA	When this bit is set to 1, the interrupt is enabled. When this bit is 0, any interrupt will be ignored. This bit can also be set by writing the “COM CPU Interrupt Enable Register (com_cpu_interrupt_enable, Offset 0x105028)” on page 870.	RW	0	X	X	X	X
6	MSG_BIT1	This is a simple RW bit.	RW	0x1	X	X	X	X
5–4	RESERVED		RO	0	X	X	X	X

Table 392: COM CPU Mode Register (com_cpu_mode, Offset 0x105000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
3	PAGE_0_INST_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_INST_HALTED bit of the “COM CPU Interrupt Enable Register (com_cpu_interrupt_enable, Offset 0x105028)” on page 870 when an instruction references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
2	PAGE_0_DATA_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_DATA_HALTED bit of the “COM CPU Interrupt Enable Register (com_cpu_interrupt_enable, Offset 0x105028)” on page 870 when data references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
1	STEP_ENA	When this bit is set, the processor is allowed to execute one cycle regardless of any halt conditions. If the halting condition still exists, the CPU will halt again after the one cycle, otherwise, it will resume normal operation.	RW	0	X	X	X	X
0	LOCAL_RST	When this bit is written to a 1, the processor will reset as if from power-up state. All Reset value of registers will be assigned.	RW	0	X	X	X	X

COM CPU STATE REGISTER (COM_CPU_STATE, OFFSET 0x105004)**Table 393: COM CPU State Register (com_cpu_state, Offset 0x105004)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BLOCKED_READ	This bit indicates that a blocking data cache miss occurred, causing the CPU to stall while data is fetched from memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred. Writing a 1 clears this bit.	WC	0	X	X	X	X
30–16	RESERVED		RO	0	X	X	X	X
15	INST_FETCH_STALL	This bit is set while the processor is stalled due to instruction fetch.	RO	0	X	X	X	X
14	DATA_ACCESS_STALL	This bit is set while the processor is stalled due to data access.	RO	0	X	X	X	X
13	RESERVED		RO	0	X	X	X	X

Table 393: COM CPU State Register (com_cpu_state, Offset 0x105004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	INTERRRUPT	This bit is set each time an interrupt input is asserted, regardless of the interrupt enable bit (see the INTERRUPT_ENA field of the "COM CPU Mode Register (com_cpu_mode, Offset 0x105000)" on page 542).	WC	0	X	X	X	X
11	SPAD_UNDERFLOW	This bit is each time an attempt is made to access the underflow area of the Scratchpad.	WC	0	X	X	X	X
10	SOFT_HALTED	This bit is set while the processor is halted due to the setting of the SOFT_HALT bit of the "COM CPU Mode Register (com_cpu_mode, Offset 0x105000)" on page 542.	RO	0	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_HALTED	This bit is set while the processor is halted due to the generation of an abort condition by one or more Fast I/O devices within the CPU block. This will only happen if halt is enabled by the FIO_ABORT_HALT_ENA bit of the "COM CPU Mode Register (com_cpu_mode, Offset 0x105000)" on page 542.	RO	0	X	X	X	X
7	ALIGN_HALTED	This bit is set while the processor is halted due to bad memory alignment problem on a load or store instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
6	BAD_PC_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
4	PAGE_0_INST_HALTED	This bit is set while the processor is halted due to executing an instruction within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
3	PAGE_0_DATA_HALTED	This bit is set while the processor is halted due to accessing data within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
2	BAD_INST_HALTED	This bit is set while the processor is halted due fetching an invalid instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT	This bit is set while the processor is halted due reaching a hardware breakpoint as enabled in the mode register. Writing a 1 clears this bit.	WC	0	X	X	X	X

COM CPU EVENT MASK REGISTER (COM_CPU_EVENT_MASK, OFFSET 0x105008)

This register provides one bit for each “[COM CPU State Register \(com_cpu_state, Offset 0x105004\)](#)” on page 543 bit to enable it into the equation for generation the TX Processor Attention output. The reset value of 1 masks all halt conditions from generating an attention.

Table 394: COM CPU Event Mask Register (com_cpu_event_mask, Offset 0x105008)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–13	RESERVED		RO	0	X	X	X	X
12	INTERRUPT_MASK	—	RW	0	X	X	X	X
11	SPAD_UNDERFLOW_MASK	—	RW	0	X	X	X	X
10	SOFT_HALTED_MASK	—	RW	0x1	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_MASK	—	RW	0x1	X	X	X	X
7	ALIGN_HALTED_MASK	—	RW	0	X	X	X	X
6	BAD_PC_HALTED_MASK	—	RW	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED_MASK	—	RW	0	X	X	X	X
4	PAGE_0_INST_HALTED_MASK	—	RW	0	X	X	X	X
3	PAGE_0_DATA_HALTED_MASK	—	RW	0	X	X	X	X
2	BAD_INST_HALTED_MASK	—	RW	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT_MASK	—	RW	0	X	X	X	X

COM CPU PROGRAM COUNTER REGISTER (COM_CPU_PROGRAM_COUNTER, OFFSET 0x10501C)

Description	Mode	Reset	06	08	09	16
This register allows the program counter to read at any time. The value can be modified RW when the processor is halted. Writes will also clear any pending instruction in the decode stage of the pipeline. Bits 31-2 are implemented. 1s written to bits 1-0 are ignored. If the processor has a ROM, then the reset value of this register points to the internal ROM. If the processor does not have a ROM, then this reset value points to the scratchpad area.	0		X	X	X	X

COM CPU REGISTER FILE REGISTERS (COM_CPU_REG_FILE[32], OFFSET 0x105200)

Description	Mode	Reset	06	08	09	16
While the processor is halted, the general-purpose processor registers (r0-r31) can be RW read and written through these register locations.	0		X	X	X	X

COM COMXQ FTQ DATA REGISTERS (COM_COMXQ_FTQ_DATA[14], OFFSET 0x105340)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	0		X	X	X	X

COM COMXQ FTQ COMMAND REGISTER (com_comxq_ftq_cmd, Offset 0x105378)

The Completion Transmit FTQ is 16 records deep.

Table 395: COM COMXQ FTQ Command Register (com_comxq_ftq_cmd, Offset 0x105378)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “ COM CPU Mode Register (com_cpu_mode, Offset 0x105000) ” on page 542. Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “ COM CPU Mode Register (com_cpu_mode, Offset 0x105000) ” on page 542. Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the “ COM CPU Mode Register (com_cpu_mode, Offset 0x105000) ” on page 542 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the “ COM CPU Mode Register (com_cpu_mode, Offset 0x105000) ” on page 542 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “ COM CPU Mode Register (com_cpu_mode, Offset 0x105000) ” on page 542 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 395: COM COMXQ FTQ Command Register (com_comxq_ftq_cmd, Offset 0x105378) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16						
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X						
24–11	RESERVED		RO	0	X	X	X	X						
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “COM CPU Mode Register (com_cpu_mode, Offset 0x105000)” on page 542.	RW	0	X	X	X	X						
Value Name Description														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">0</td> <td style="width: 10%;">0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </table>									0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.												
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.												
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “COM CPU Mode Register (com_cpu_mode, Offset 0x105000)” on page 542.	RW	0	X	X	X	X						

COM COMXQ FTQ CONTROL REGISTER (COM_COMXQ_FTQ_CTL, OFFSET 0x10537c)*Table 396: COM COMXQ FTQ Control Register (com_comxq_ftq_ctl, Offset 0x10537c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

COM COMTQ FTQ DATA REGISTERS (COM_COMTQ_FTQ_DATA[14], OFFSET 0x105380)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this block.	RO	0	X	X	X	X

COM COMTQ FTQ COMMAND REGISTER (COM_COMTQ_FTQ_CMD, OFFSET 0x1053B8)

The Completion Timer FTQ is 32 records deep.

Table 397: COM COMTQ FTQ Command Register (com_comtq_ftq_cmd, Offset 0x1053b8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "COM COMTQ FTQ Control Register (com_comtq_ftq_ctl, Offset 0x1053bc)" on page 552 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "COM COMTQ FTQ Control Register (com_comtq_ftq_ctl, Offset 0x1053bc)" on page 552 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the "COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)" on page 552 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the "COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)" on page 552 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the "COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)" on page 552 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 397: COM COMTQ FTQ Command Register (com_comtq_ftq_cmd, Offset 0x1053b8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16	
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X	
24–11	RESERVED		RO	0	X	X	X	X	
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)” on page 552.	RW	0	X	X	X	X	
Value Name Description									
		0 0 When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.							
		1 1 When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.							
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc)” on page 555.	RW	0	X	X	X	X	

COM COMTQ FTQ CONTROL REGISTER (COM_COMTQ_FTQ_CTL, OFFSET 0x1053BC)*Table 398: COM COMTQ FTQ Control Register (com_comtq_ftq_ctl, Offset 0x1053bc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is set, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (see ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

COM COMQ FTQ DATA REGISTERS (COM_COMQ_FTQ_DATA[14], OFFSET 0x1053C0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this block.	RO	0	X	X	X	X

COM COMQ FTQ COMMAND REGISTER (COM_COMQ_FTQ_CMD, OFFSET 0x1053F8)

The main completion FTQ is filled by the RX DMA. This FTQ is unique in that it provides out-of-order writes from the RX DMA. This feature is enabled by bit 3 of the **ctl** register below. When this mode is enabled, GRC writes of the FTQ will not be executed as the re-order processing prevents insertion of these extra records. The Completion FTQ is 16 records deep.

Table 399: COM COMQ FTQ Command Register (com_comq_ftq_cmd, Offset 0x1053f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc)” on page 555 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc)” on page 555 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the “COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)” on page 552 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the “COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)” on page 552 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0)” on page 552 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 399: COM COMQ FTQ Command Register (com_comq_ftq_cmd, Offset 0x1053f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16								
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X								
24–11	RESERVED		RO	0	X	X	X	X								
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “ COM COMQ FTQ Data Registers (com_comq_ftq_data[14], Offset 0x1053c0) ” on page 552.	RW	0	X	X	X	X								
		<table border="1"> <thead> <tr> <th>Val</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </tbody> </table>	Val	Name	Description	0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.					
Val	Name	Description														
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.														
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.														
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “ COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc) ” on page 555.	RW	0	X	X	X	X								

COM COMQ FTQ CONTROL REGISTER (COM_COMQ_FTQ_CTL, OFFSET 0x1053FC)*Table 400: COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

COM SCRATCHPAD AREA (COM_SCRATCH[8192], OFFSET 0x120000)

Description	Mode	Reset	06	08	09	16
This is the COM scratchpad area which is visible at 0x0 by the COM processor. RW This read/write area can be modified at any time and may be used for processor-to-processor communication.		0	X	—	—	—

COM SCRATCHPAD AREA (COM_SCRATCH[10240], OFFSET 0x120000)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
This is the COM scratchpad area which is visible at 0x0 by the COM processor. RW This read/write area can be modified at any time and may be used for processor-to-processor communication.	0	-	X	X	X	

MANAGEMENT CONTROL PROCESSOR (MCP) REGISTERS

MCP CPU MODE REGISTER (MCP_CPU_MODE, OFFSET 0x145000)

Table 401: MCP CPU Mode Register (mcp_cpu_mode, Offset 0x145000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	SPAD_UNDERFLOW_HALT_ENA	When this bit is set, the CPU will halt when the SPAD_UNDERFLOW bit of the “ COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc) ” on page 555 is set.	RW	0x1	X	X	X	X
14	RESERVED		RO	0	X	X	X	X
13	FIO_ABORT_HALT_ENA	When this bit is set, the CPU will halt when an abort is indicated from any Fast I/O space peripheral.	RW	0x1	X	X	X	X
12	BAD_INST_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_PC_HALTED field of the “ COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc) ” on page 555 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
11	BAD_DATA_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_DATA_ADDR_HALTED bit in the “ COM COMQ FTQ Control Register (com_comq_ftq_ctl, Offset 0x1053fc) ” on page 555 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
10	SOFT_HALT	When this bit is set, the CPU will halt. This bit is cleared by an exception or reset. If the processor does not have a ROM, then this bit will reset to set so that no code is executed from the scratchpad. If the processor does have a ROM, this bit resets a cleared so that the processor executes from ROM after reset.	RW	0	X	X	X	X
9–8	RESERVED		RO	0	X	X	X	X
7	INTERRUPT_ENA	When this bit is set to 1, the interrupt is enabled. When this bit is 0, any interrupt will be ignored. This bit can also be set by writing the “ MCP CPU Interrupt Enable Register (mcp_cpu_interrupt_enable, Offset 0x145028) ” on page 887.	RW	0	X	X	X	X
6	MSG_BIT1	This is a simple RW bit.	RW	0x1	X	X	X	X
5–4	RESERVED		RO	0	X	X	X	X

Table 401: MCP CPU Mode Register (mcp_cpu_mode, Offset 0x145000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
3	PAGE_0_INST_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_INST_HALTED bit of the “MCP CPU Interrupt Enable Register (mcp_cpu_interrupt_enable, Offset 0x145028)” on page 887 when an instruction references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
2	PAGE_0_DATA_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_DATA_HALTED bit of the “MCP CPU Interrupt Enable Register (mcp_cpu_interrupt_enable, Offset 0x145028)” on page 887 when data references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
1	STEP_ENA	When this bit is set, the processor is allowed to execute one cycle regardless of any halt conditions. If the halting condition still exists, the CPU will halt again after the one cycle, otherwise, it will resume normal operation.	RW	0	X	X	X	X
0	LOCAL_RST	When this bit is written to a 1, the processor will reset as if from power-up state. All Reset value of registers will be assigned.	RW	0	X	X	X	X

MCP CPU STATE REGISTER (MCP_CPU_STATE, OFFSET 0x145004)**Table 402: MCP CPU State Register (mcp_cpu_state, Offset 0x145004)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BLOCKED_READ	This bit indicates that a blocking data cache miss occurred, causing the CPU to stall while data is fetched from memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred. Writing a 1 clears this bit.	WC	0	X	X	X	X
30–16	RESERVED		RO	0	X	X	X	X
15	INST_FETCH_STALL	This bit is set while the processor is stalled due to instruction fetch.	RO	0	X	X	X	X
14	DATA_ACCESS_STALL	This bit is set while the processor is stalled due to data access.	RO	0	X	X	X	X
13	RESERVED		RO	0	X	X	X	X

Table 402: MCP CPU State Register (*mcp_cpu_state*, Offset 0x145004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	INTERRRUPT	This bit is set each time an interrupt input is asserted, regardless of the interrupt enable bit (see the INTERRUPT_ENA field of the “MCP CPU Mode Register (<i>mcp_cpu_mode</i> , Offset 0x145000)” on page 557).	WC	0	X	X	X	X
11	SPAD_UNDERFLOW	This bit is each time an attempt is made to access the underflow area of the Scratchpad.	WC	0	X	X	X	X
10	SOFT_HALTED	This bit is set while the processor is halted due to the setting of bit 10 in the mode register.	RO	0	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_HALTED	This bit is set while the processor is halted due to the generation of an abort condition by one or more Fast I/O devices within the CPU block. This will only happen if halt is enabled by the FIO_ABORT_HALT_ENA bit of the “MCP CPU Mode Register (<i>mcp_cpu_mode</i> , Offset 0x145000)” on page 557.	RO	0	X	X	X	X
7	ALIGN_HALTED	This bit is set while the processor is halted due to bad memory alignment problem on a load or store instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
6	BAD_PC_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
4	PAGE_0_INST_HALTED	This bit is set while the processor is halted due to executing an instruction within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
3	PAGE_0_DATA_HALTED	This bit is set while the processor is halted due to accessing data within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
2	BAD_INST_HALTED	This bit is set while the processor is halted due fetching an invalid instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT	This bit is set while the processor is halted due reaching a hardware breakpoint as enabled in the mode register. Writing a 1 clears this bit.	WC	0	X	X	X	X

MCP CPU EVENT MASK REGISTER (MCP_CPU_EVENT_MASK, OFFSET 0x145008)

This register provides one bit for each “[MCP CPU State Register \(mcp_cpu_state, Offset 0x145004\)](#)” on page 558 bit to enable it into the equation for generation the TX Processor Attention output. The reset value of 1 masks all halt conditions from generating an attention.

Table 403: MCP CPU Event Mask register (mcp_cpu_event_mask, Offset 0x145008)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–13	RESERVED		RO	0	X	X	X	X
12	INTERRUPT_MASK	—	RW	0	X	X	X	X
11	SPAD_UNDERFLOW_MASK	—	RW	0	X	X	X	X
10	SOFT_HALTED_MASK	—	RW	0x1	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_MASK	—	RW	0x1	X	X	X	X
7	ALIGN_HALTED_MASK	—	RW	0	X	X	X	X
6	BAD_PC_HALTED_MASK	—	RW	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED_MASK	—	RW	0	X	X	X	X
4	PAGE_0_INST_HALTED_MASK	—	RW	0	X	X	X	X
3	PAGE_0_DATA_HALTED_MASK	—	RW	0	X	X	X	X
2	BAD_INST_HALTED_MASK	—	RW	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT_MASK	—	RW	0	X	X	X	X

MCP CPU PROGRAM COUNTER REGISTER (MCP_CPU_PROGRAM_COUNTER, OFFSET 0x14501c)

Description	Mode	Reset	06	08	09	16
This register allows the program counter to read at any time. The value can be modified RW when the processor is halted. Writes will also clear any pending instruction in the decode stage of the pipeline. Bits 31-2 are implemented. 1s written to bits 1-0 are ignored. If the processor has a ROM, then the reset value of this register points to the internal ROM. If the processor does not have a ROM, then this reset value points to the scratchpad area.	0		X	X	X	X

MCP CPU REGISTER FILE REGISTERS (MCP_CPU_REG_FILE[32], OFFSET 0x145200)

Description	Mode	Reset	06	08	09	16
While the processor is halted, the general-purpose processor registers (r0–r31) can be RW read and written through these register locations.	0		X	X	X	X

MCP MCPQ FTQ DATA REGISTERS (MCP_MCPQ_FTQ_DATA[14], OFFSET 0x1453c0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	0	X X X X				

MCP MCPQ FTQ COMMAND REGISTER (MCP_MCPQ_FTQ_CMD, OFFSET 0x1453f8)

The Management Processor FTQ is 32 records deep.

Table 404: MCP MCPQ FTQ Command Register (mcp_mcpq_ftq_cmd, Offset 0x1453f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X X X X			
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "MCP MCPQ FTQ Control Register (mcp_mcpq_ftq_ctl, Offset 0x1453fc)" on page 563 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X X X X			
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "MCP MCPQ FTQ Control Register (mcp_mcpq_ftq_ctl, Offset 0x1453fc)" on page 563 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X X X X			
28	ADD_DATA	When this bit is written as a 1, the "MCP MCPQ FTQ Data Registers (mcp_mcpq_ftq_data[14], Offset 0x1453c0)" on page 561 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X X X X			
27	ADD_INTERVENE	When this bit is written as a 1, the "MCP MCPQ FTQ Data Registers (mcp_mcpq_ftq_data[14], Offset 0x1453c0)" on page 561 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X X X X			

Table 404: MCP MCPQ FTQ Command Register (mcp_mcpq_ftq_cmd, Offset 0x1453f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16						
26	RD_DATA	When this bit is written as a 1, the “ MCP MCPQ FTQ Data Registers (mcp_mcpq_ftq_data[14], Offset 0x1453c0) ” on page 561 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X						
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X						
24–11	RESERVED		RO	0	X	X	X	X						
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “ MCP MCPQ FTQ Data Registers (mcp_mcpq_ftq_data[14], Offset 0x1453c0) ” on page 561.	RW	0	X	X	X	X						
Val Name Description														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">0</td> <td style="width: 10%;">0</td> <td>When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.</td> </tr> <tr> <td>1</td> <td>1</td> <td>When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.</td> </tr> </table>									0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.	1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.
0	0	When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.												
1	1	When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.												
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “ MCP MCPQ FTQ Control Register (mcp_mcpq_ftq_ctl, Offset 0x1453fc) ” on page 563.	RW	0	X	X	X	X						

MCP MCPQ FTQ CONTROL REGISTER (MCP_MCPQ_FTQ_CTL, OFFSET 0x1453FC)*Table 405: MCP MCPQ FTQ Control Register (mcp_mcpq_ftq_ctl, Offset 0x1453fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the overflow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

MCP SCRATCHPAD AREA (MCP_SCRATCH[8192], OFFSET 0x1600000)

Description	Mode	Reset	06	08	09	16
This is the MCP scratch pad space which is visible at 0x0 by the MCP processor. This RO can be modified at any time and may be used for processor-to-processor communication.	RO	0	X	X	—	—

MCP SCRATCHPAD AREA (MCP_SCRATCH[16384], OFFSET 0x160000)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
This is the MCP scratch pad space which is visible at 0x0 by the MCP processor. This RO can be modified at any time and may be used for processor-to-processor communication.	0	-	-	X	X	

COMMAND PROCESSOR (COM) REGISTERS

CP CPU MODE REGISTER (CP_CPU_MODE, OFFSET 0x185000)

Table 406: CP CPU Mode Register (cp_cpu_mode, Offset 0x185000)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	X	X
15	SPAD_UNDERFLOW_HALT_ENA	When this bit is set, the CPU will halt when the SPAD_UNDERFLOW bit of the “CP CPU State Register (cp_cpu_state, Offset 0x185004)” on page 894 is set.	RW	0x1	X	X	X	X
14	RESERVED		RO	0	X	X	X	X
13	FIO_ABORT_HALT_ENA	When this bit is set, the CPU will halt when a abort is indicated from any Fast I/O space peripheral.	RW	0x1	X	X	X	X
12	BAD_INST_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_PC_HALTED field of the “CP CPU State Register (cp_cpu_state, Offset 0x185004)” on page 894 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
11	BAD_DATA_HALT_ENA	When this bit is set, the CPU will halt when any condition that causes the BAD_DATA_ADDR_HALTED bit in the “CP CPU State Register (cp_cpu_state, Offset 0x185004)” on page 894 to be set occurs. This bit is cleared by an interrupt.	RW	0x1	X	X	X	X
10	SOFT_HALT	When this bit is set, the CPU will halt. This bit is cleared by an exception or reset. If the processor does not have a ROM, then this bit will reset to set so that no code is executed from the scratchpad. If the processor does have a ROM, this bit resets a cleared so that the processor executes from ROM after reset.	RW	0	X	X	X	X
9–8	RESERVED		RO	0	X	X	X	X
7	INTERRUPT_ENA	When this bit is set to 1, the interrupt is enabled. When this bit is 0, any interrupt will be ignored. This bit can also be set by writing the “CP CPU Interrupt Enable Register (cp_cpu_interrupt_enable, Offset 0x185028)” on page 897.	RW	0	X	X	X	X
6	MSG_BIT1	This is a simple RW bit.	RW	0x1	X	X	X	X
5–4	RESERVED		RO	0	X	X	X	X

Table 406: CP CPU Mode Register (cp_cpu_mode, Offset 0x185000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
3	PAGE_0_INST_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_INST_HALTED bit of the “CP CPU State Register (cp_cpu_state, Offset 0x185004)” on page 894 when an instruction references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
2	PAGE_0_DATA_ENA	This bit enables the processor to halt and to latch the value of the PAGE_0_DATA_HALTED bit of the “CP CPU State Register (cp_cpu_state, Offset 0x185004)” on page 894 when data references the first 256 bytes of memory space (page 0). This bit is cleared by an interrupt or reset.	RW	0	X	X	X	X
1	STEP_ENA	When this bit is set, the processor is allowed to execute one cycle regardless of any halt conditions. If the halting condition still exists, the CPU will halt again after the one cycle, otherwise, it will resume normal operation.	RW	0	X	X	X	X
0	LOCAL_RST	When this bit is written to a 1, the processor will reset as if from power-up state. All Reset value of registers will be assigned.	RW	0	X	X	X	X

CP CPU STATE REGISTER (CP_CPU_STATE, OFFSET 0x185004)**Table 407: CP CPU State Register (cp_cpu_state, Offset 0x185004)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BLOCKED_READ	This bit indicates that a blocking data cache miss occurred, causing the CPU to stall while data is fetched from memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred. Writing a 1 clears this bit.	WC	0	X	X	X	X
30–16	RESERVED		RO	0	X	X	X	X
15	INST_FETCH_STALL	This bit is set while the processor is stalled due to instruction fetch.	RO	0	X	X	X	X
14	DATA_ACCESS_STALL	This bit is set while the processor is stalled due to data access.	RO	0	X	X	X	X
13	RESERVED		RO	0	X	X	X	X

Table 407: CP CPU State Register (cp_cpu_state, Offset 0x185004) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
12	INTERRRUPT	This bit is set each time an interrupt input is asserted, regardless of the interrupt enable bit (see the INTERRUPT_ENA field of the “CP CPU Mode Register (cp_cpu_mode, Offset 0x185000)” on page 565).	WC	0	X	X	X	X
11	SPAD_UNDERFLOW	This bit is each time an attempt is made to access the underflow area of the Scratchpad.	WC	0	X	X	X	X
10	SOFT_HALTED	This bit is set while the processor is halted due to the setting of the SOFT_HALT bit of the “CP CPU Mode Register (cp_cpu_mode, Offset 0x185000)” on page 565.	RO	0	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_HALTED	This bit is set while the processor is halted due to the generation of an abort condition by one or more Fast IO devices within the CPU block. This will only happen if halt is enabled by the FIO_ABORT_HALT_ENA bit of the “CP CPU Mode Register (cp_cpu_mode, Offset 0x185000)” on page 565.	RO	0	X	X	X	X
7	ALIGN_HALTED	This bit is set while the processor is halted due to bad memory alignment problem on a load or store instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
6	BAD_PC_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED	This bit is set while the processor is halted due to bad value in the Program Counter (PC). Writing a 1 clears this bit.	WC	0	X	X	X	X
4	PAGE_0_INST_HALTED	This bit is set while the processor is halted due to executing an instruction within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
3	PAGE_0_DATA_HALTED	This bit is set while the processor is halted due to accessing data within page 0 (the first 256 bytes) of memory. Writing a 1 clears this bit.	WC	0	X	X	X	X
2	BAD_INST_HALTED	This bit is set while the processor is halted due fetching an invalid instruction. Writing a 1 clears this bit.	WC	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT	This bit is set while the processor is halted due reaching a hardware breakpoint as enabled in the mode register. Writing a 1 clears this bit.	WC	0	X	X	X	X

CP CPU EVENT MASK REGISTER (CP_CPU_EVENT_MASK, OFFSET 0x185008)

This register provides one bit for each “[CP CPU State Register \(cp_cpu_state, Offset 0x185004\)](#)” on page 566 bit to enable it into the equation for generation the TX Processor Attention output. The reset value of 1 masks all halt conditions from generating an attention.

Table 408: CP CPU Event Mask Register (cp_cpu_event_mask, Offset 0x185008)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–13	RESERVED		RO	0	X	X	X	X
12	INTERRUPT_MASK	—	RW	0	X	X	X	X
11	SPAD_UNDERFLOW_MASK	—	RW	0	X	X	X	X
10	SOFT_HALTED_MASK	—	RW	0x1	X	X	X	X
9	RESERVED		RO	0	X	X	X	X
8	FIO_ABORT_MASK	—	RW	0x1	X	X	X	X
7	ALIGN_HALTED_MASK	—	RW	0	X	X	X	X
6	BAD_PC_HALTED_MASK	—	RW	0	X	X	X	X
5	BAD_DATA_ADDR_HALTED_MASK	—	RW	0	X	X	X	X
4	PAGE_0_INST_HALTED_MASK	—	RW	0	X	X	X	X
3	PAGE_0_DATA_HALTED_MASK	—	RW	0	X	X	X	X
2	BAD_INST_HALTED_MASK	—	RW	0	X	X	X	X
1	RESERVED		RO	0	X	X	X	X
0	BREAKPOINT_MASK	—	RW	0	X	X	X	X

CP CPU PROGRAM COUNTER REGISTER (CP_CPU_PROGRAM_COUNTER, OFFSET 0x18501c)

Description	Mode	Reset	06	08	09	16
This register allows the program counter to be read at any time. The value can be RW modified when the processor is halted. Writes will also clear any pending instruction in the decode stage of the pipeline. Bits 31-2 are implemented. 1s written to bits 1-0 are ignored. If the processor has a ROM, then the reset value of this register points to the internal ROM. If the processor does not have a ROM, then this reset value points to the scratchpad area.	0		X	X	X	X

CP CPU REGISTER FILE REGISTERS (CP_CPU_REG_FILE[32], OFFSET 0x185200)

Description	Mode	Reset	06	08	09	16
While the processor is halted, the general-purpose processor registers (r0–r31) can be RW read and written through these register locations.	0		X	X	X	X

CP CPQ FTQ DATA REGISTERS (CP_CPQ_FTQ_DATA[14], OFFSET 0x1853c0)

<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
This area is used to access the FTQ data in the holding register within the FTQ for this RO block.	0	X X X X				

CP CPQ FTQ COMMAND REGISTER (CP_CPQ_FTQ_CMD, OFFSET 0x1853f8)

The Command Processor FTQ is 8 records deep.

Table 409: CP CPQ FTQ Command Register (cp_cpq_ftq_cmd, Offset 0x1853f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the “CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc)” on page 571 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the “CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc)” on page 571 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the “CP CPQ FTQ Data Registers (cp_cpq_ftq_data[14], Offset 0x1853c0)” on page 899 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the “CP CPQ FTQ Data Registers (cp_cpq_ftq_data[14], Offset 0x1853c0)” on page 899 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the “CP CPQ FTQ Data Registers (cp_cpq_ftq_data[14], Offset 0x1853c0)” on page 899 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 409: CP CPQ FTQ Command Register (cp_cpq_ftq_cmd, Offset 0x1853f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the “ CP CPQ FTQ Data Registers (cp_cpq_ftq_data[14], Offset 0x1853c0) ” on page 899.	RW	0	X	X	X	X
		Value Name Description						
		0 0 When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.						
		1 1 When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.						
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the Offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “ CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc) ” on page 571.	RW	0	X	X	X	X

CP CPQ FTQ CONTROL REGISTER (CP_CPQ_FTQ_CTL, OFFSET 0x1853FC)**Table 410: CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc)**

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X

Table 410: CP CPQ FTQ Control Register (cp_cpq_ftq_ctl, Offset 0x1853fc) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

CP SCRATCHPAD AREA (CP_SCRATCH[8192], OFFSET 0x1A0000)

Description	Mode	Reset	06	08	09	16
This is the CP scratchpad area which is visible at 0x0 by the CP processor. This read/ RW write area can be modified at any time and may be used for processor-to-processor communication.	0	X X X X				

CP SCRATCHPAD AREA (CP_SCRATCH[10240], OFFSET 0x1A0000)

Description	Mode	Reset	06	08	09	16
This is the CP scratchpad area which is visible at 0x0 by the CP processor. This read/ RW write area can be modified at any time and may be used for processor-to-processor communication.	0	X X X X				

TRANSMIT ASSEMBLER (TAS) REGISTERS

The Transmit Assembler (TAS) processor reads data from the header and payload queues and combines it into a 16-bit wide Transmit MAC data stream. The Transmit Assembler also houses the Transmit Header and Transmit Payload Queues. These are arbitrated memories that are used in a circular queue fashion to store data from the Transmit Processor (Header) and the Transmit DMA (Payload).

TAS COMMAND REGISTER (TAS_COMMAND, OFFSET 0x1c0000)

Table 411: TAS Command Register (tas_command, Offset 0x1c0000)

Bit	Name	Description	Mode	Reset	06	06	09	16
31	RESET_STATE	When this bit is written as a 1, the TX Assembler is placed into reset state. The bit must be written back to 0 to allow the TX Assembler to operate again.	RW	0	X	X	X	X
30–8	RESERVED		RO	0	X	X	X	X
7	TPBUF_INIT	When this bit is written as a 1, the TX Payload Buffer memory will be written to zeroes to allow parity checking to be enabled. While this clear occurs, the parity error output for the TX Payload Buffer memory will be masked.	SC	0	X	X	X	X
6	THBUF_INIT	When this bit is written as a 1, the TX Header Buffer memory will be written to zeroes to allow parity checking to be enabled. While this clear is happening, the parity error output for the TX Header Buffer memory will be masked.	SC	0	X	X	X	X
5	PKT_END_TOSHORT_ABORT	This bit is set when the TX Assembler processes a record when the PKT_END field is not set and was set on the previous record and the record length is less than 60 bytes. This means that there was not enough data in the first record of a frame to handle the re-transmit if a legal collision was received by the MAC. Short single record packets are permissible and should be passed and retransmitted without error. The TX Assembler will stop processing packets until this bit is cleared.	WC	0	X	X	X	X
4–3	RESERVED		RO	0	X	X	X	X
2	THBUF_ENABLE	This bit is read as a 1 when the TX_HEADER_Q_ENABLE field is enabled. Writing this bit has no effect.	RO	0	X	X	X	X
1	TPBUF_ENABLE	This bit is read as a 1 when the TX_PAYLOAD_Q_ENABLE field is enabled. Writing this bit has no effect.	RO	0	X	X	X	X

Table 411: TAS Command Register (tas_command, Offset 0x1c0000) (Cont.)

Bit	Name	Description	Mode	Reset	06	06	09	16
0	TAS_ENABLE	This bit is read as a 1 when the TX_ASSEMBLER_ENABLE field is enabled. Writing this bit has no effect.	RO	0	X	X	X	X

TAS FTQ DATA REGISTERS (TAS_FTQ_DATA[14], OFFSET 0x1c03c0)

Description	Mode	Reset	06	08	09	16
This area is used to access the FTQ data in the holding register within the FTQ for this block.	RO	0	X	X	X	X

TAS FTQ COMMAND REGISTER (TAS_FTQ_CMD, OFFSET 0x1c03f8)

The TX Assembler FTQ is 16 records deep.

Table 412: TAS FTQ Command Register (tas_ftq_cmd, Offset 0x1c03f8)

Bit	Name	Description	Mode	Reset	06	08	09	16
31	BUSY	This bit reads as 1 when the FTQ controller is busy and cannot respond to any of the command bits below. This bit must be 0 when POP, INTERVENE_CLR, and ADD command bits below are written as 1.	RO	0	X	X	X	X
30	POP	When this bit is written as a 1, the top value of the FTQ is popped (removed). This should only be done if the top value has the Intervene bit set as indicated in the "TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)" on page 578 . Writing this bit as a 0 has no effect. The BUSY bit must be 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
29	INTERVENE_CLR	When this bit is written as a 1, the INTERVENE bit in the top value of the FTQ is cleared. This should only be done if the top value has the INTERVENE bit set as indicated in the "TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)" on page 578 . Writing this bit as a 0 has no effect. Clearing the INTERVENE bit will allow the value to be processed by the TX DMA block normally. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as 0.	SC	0	X	X	X	X
28	ADD_DATA	When this bit is written as a 1, the "TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)" on page 578 is written to the bottom of the FTQ. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
27	ADD_INTERVEN	When this bit is written as a 1, the "TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)" on page 578 is written to the bottom of the FTQ just as when the ADD_DATA bit is set, except that the INTERVENE bit is set on the entry. The WR_TOP bit must be cleared for this to work. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X
26	RD_DATA	When this bit is written as a 1, the "TAS FTQ Data Registers (tas_ftq_data[14], Offset 0x1c03c0)" on page 907 is updated with the data from the FTQ entry pointed to by the OFFSET register. The BUSY bit will remain set until the read is complete and the data area can be read. Writing this bit as a 0 has no effect. The BUSY bit must be a 0 when this bit is written to a 1. This bit always reads as a 0.	SC	0	X	X	X	X

Table 412: TAS FTQ Command Register (*tas_ftq_cmd*, Offset 0x1c03f8) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
25	SFT_RESET	When this bit is written as a 1, the FTQ block will be reset. All data in the FTQ will be lost. This bit is intended for diagnostic use only. Setting this bit may cause down stream state machines to be confused because the Valid output may be deasserted unexpectedly.	SC	0	X	X	X	X
24–11	RESERVED		RO	0	X	X	X	X
10	WR_TOP	This bit controls the operation of the exchanges between the FTQ and the TAS FTQData Registers: see “TAS FTQ Data Registers (<i>tas_ftq_data[14]</i> , Offset 0x1c03c0)” on page 907.	RW	0	X	X	X	X
		Val Name Description						
		0 0 When 0, RD_DATA commands, will read the top entry in the FTQ to the holding register. ADD_DATA and ADD_INTERVENE commands will write the holding register to the bottom of the FTQ.						
		1 1 When 1, both reads and writes of the holding register will access the entry as specified in the OFFSET value. This should only be done while the chip is idle or an INTERVENE record is at the top of the queue. This mode is used for either modifying the top of the queue or as a debug tool for reading the complete queue.						
9–0	OFFSET	This value is used when the WR_TOP bit is set to 1 and controls the offset into the FTQ that is being accessed for transfers between the FTQ and the holding register. When this value is 0, the top of the queue entry will be accessed. When this value is 1, the top-1 value is being accessed. This value must never be programmed larger than the default value of the MAX_DEPTH field of the “TAS FTQ Control Register (<i>tas_ftq_ctl</i> , Offset 0x1c03fc)” on page 578.	RW	0	X	X	X	X

TAS FTQ CONTROL REGISTER (TAS_FTQ_CTL, OFFSET 0x1C03FC)*Table 413: TAS FTQ Control Register (tas_ftq_ctl, Offset 0x1c03fc)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–22	CUR_DEPTH	This value indicates the current FTQ depth being used. A value of 0 indicates that the FTQ is empty.	RO	0	X	X	X	X
21–12	MAX_DEPTH	This value controls the amount of the FTQ that will be available and essentially shortens the queue depth. The default value is for maximum queue depth. There are two reasons to shorten the queue. First, it can decrease total chip latency. Second, it reserves some space in the queue so that old entries can be viewed after a hang. This is because they will not be written over as the system invariably backs-up. The default value of this register is the maximum number of entries that will fit into the FTQ.	RW	X	X	X	X	X
11–3	RESERVED		RO	0	X	X	X	X
2	FORCE_INTERVENE	When this bit is 1, it will force the intervene bit to be set on each entry written to the FTQ by any of the hardware interfaces. The register control of the INTERVENE bit on writes still works normally (See ADD_INTERVENE). When this bit is 0, the INTERVENE input to the hardware interfaces works normally.	RW	0	X	X	X	X
1	OVERFLOW	When this bit is set, it indicates that a request was made to write the FTQ when it was at its programmed maximum depth. This is to indicate that the chip was backed-up in this area at some time since the bit was last cleared. Writing this bit to a 1 will clear the over-flow status. Writing this bit as a 0 has no effect.	WC	0	X	X	X	X
0	INTERVENE	This bit indicates the intervene status of the top entry in the FTQ. If set, it indicates that the top entry has the INTERVENE bit set and requires firmware intervention.	RO	0	X	X	X	X

The Receive Memory Buffer (RBUF) block provides storage for packets received from the network until they are processed and moved into host memory. The storage memory is divided into 128 byte chunks called MBUFs. Memory allocation is done on a per MBUF basis and de-allocation is done on a per-packet basis. The used MBUFs are managed using multiple linked lists on a per-requestor/packet basis. The unused MBUFs are managed using a special linked list called the free list. Based on programmable thresholds, the RBUF block provides indications about storage memory utilization which are used to either discard packets or generate flow control frames.

RBUF COMMAND REGISTER (RBUF_COMMAND, OFFSET 0x200000)*Table 414: RBUF Command Register (rbuf_command, Offset 0x200000)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–8	RESERVED		RO	0	X	X	X	X
7–6	RESERVED		RO	0	X	X	—	—
7	GRC_ENDIAN_CONV_DIS	Setting this bit disables RBUF from performing little-endian to big-endian conversion.	RW	0	—	—	X	X
6	EN_PRI_CHANGE	When this bit is set MBUF allocation will have higher priority than cluster de-allocation. When the packet buffer is full then cluster de-allocation will have higher priority.	RW	0x1	—	—	X	X
5	ALLOC_REQ	This bit causes a single MBUF to be allocated and placed in the VALUE field when it is written as a 1. The user must wait for this bit to read as 0 after setting before reading the mbuf value. This bit must be 0 before it is written as a 1.	RW	0	X	X	—	—
5	CU_ISOLATE	When this bit is set RBUF will isolate catchup traffic from regular receive traffic by reserving buffer space for each traffic type. This bit should be modified only after reset and before the RBUF block has been enabled.	RW	0	—	—	X	X
4	OVER_FREE	This bit indicates that too many buffers have been freed and the buffer count has overflowed and is a fatal condition. This bit is cleared by setting the FREE_INIT field. When this bit is set, the attention output of the RBUF block is asserted.	RO	0	X	X	X	X
3	RESERVED		RO	0	X	—	—	—
3	PKT_OFFSET_OVFL	This bit indicates whether an offset overflow attention has occurred. This means that one of the user blocks has made an access where the offset exceeds the number of MBUF blocks indicated by the cluster. This bit is set when an error occurs and is reset by a core reset. When this bit is 1 the attention output of the RX MBUF block is asserted.	RO	0	—	X	X	X

Table 414: RBUF Command Register (*rbuf_command*, Offset 0x200000) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
2	RAM_INIT	When this bit is written to 1, the RX MBUF memory contents will be cleared to 0. This will prepare the block for parity detection enable. This function does not run automatically at CORE Reset. The bit will read 1 until the operation is complete. Any requests for RX MBUF resources made while this operation is in effect will be deferred until the initialization is complete. The parity error output for the RX MBUF Block will be masked while this command is operation	SC	0	X	X	X	X
1	FREE_INIT	When this bit is written to 1, the page pointer table will be initialized such that all the buffers are free. This function is automatically executed at CORE Reset. The bit will read 1 until the operation is complete. Any requests for RX MBUF resources made while this operation is in effect will be deferred until the initialization is complete. This bit is actually reset to 1 at CORE Reset, but the free operation normally completes before the vale of 1 can be read.	SC	0	X	X	X	X
0	ENABLED	This bit indicates the current enable status of this block. If this bit is 1, it indicates that the RX_MBUF_ENABLE bit is set. Writing this bit as a 0 has no effect.	RO	0	X	X	X	X

RBUF STATUS 1 REGISTER (RBUF_STATUS1, OFFSET 0x200004)*Table 415: RBUF Status 1 Register (rbuf_status1, Offset 0x200004)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–10	RESERVED		RO	0	X	X	X	X
9–0	FREE_COUNT	This value indicates the number of MBUF blocks that are in the free list and accounts for the two buffers that are checked out by the RPM and RPC after each CORE Reset. The true hardware reset value is 0x200, but no observer will ever see that value. So the more common value that will be seen by a register read is reflected here.	RO	0x1fe	X	X	X	X

RBUF STATUS 2 REGISTER (RBUF_STATUS2, OFFSET 0x200008)*Table 416: RBUF Status 2 Register (rbuf_status2, Offset 0x200008)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–25	RESERVED		RO	0	X	X	X	X
24–16	FREE_HEAD	This value indicates the mbuf number that is at the head of the free list. This is the next MBUF that will be allocated.	RO	0	X	X	X	X
15–9	RESERVED		RO	0	X	X	X	X
8–0	FREE_TAIL	This value indicates the mbuf number that is at the tail of the free list. This is the most recently freed MBUF.	RO	0	X	X	X	X

RBUF CONFIGURATION REGISTER (RBUF_CONFIG, OFFSET 0x20000c)*Table 417: RBUF Configuration Register (rbuf_config, Offset 0x20000c)*

<i>Bit</i>	<i>Name</i>	<i>Description</i>	<i>Mode</i>	<i>Reset</i>	<i>06</i>	<i>08</i>	<i>09</i>	<i>16</i>
31–26	RESERVED		RO	0	X	X	X	X
25–16	XON_TRIP	This is the free mbuf count where the MAC will be told to send an XON pause packet. The pause output will be deasserted when the FREE_COUNT field of the RBUF Status 1 Register (see "RBUF Status 1 Register (rbuf_status1, Offset 0x200004)" on page 581) is greater than or equal to this value.	RW	0x42	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X

Table 417: RBUF Configuration Register (rbuf_config, Offset 0x20000c) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
9–0	XOFF_TRIP	This is the free mbuf count where the MAC will be told to send an XOFF pause packet and is intended to cause the link partner MAC to stop sending packets. The pause output will be asserted when the value of the FREE_COUNT field in the RBUF Status 1 Register (see “RBUF Status 1 Register (rbuf_status1, Offset 0x200004)” on page 581) drops below this value. It will stay set until the FREE_COUNT value is greater than or equal to the value in the XON_TRIP field.	RW	0x36	X	X	X	X

RBUF FIRMWARE BUFFER ALLOCATION REGISTER (RBUF_FW_BUF_ALLOC, OFFSET 0x200010)

Table 418: RBUF Firmware Buffer Allocation Register (rbuf_fw_buf_alloc, Offset 0x200010)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–16	RESERVED		RO	0	X	X	–	–
31	ALLOC_REQ	When this bit is set RBUF will allocate a single MBUF and place its address in the VALUE field.	SC	0	–	–	X	X
30–17	RESERVED		RO	0	–	–	X	X
16	TYPE	This bit determines the type of MBUF that should be allocated when the ALLOC_REQ bit is set. When this bit is set a catchup MBUF will be allocated. When this bit is clear an RX traffic MBUF will be allocated. This bit is only valid when the CU_ISOLATE bit of the RBUF Command Register (see “RBUF Command Register (rbuf_command, Offset 0x200000)” on page 579) is set.	RW	0	–	–	X	X
15–7	VALUE	This field will return an allocated MBUF after the ALLOC_REQ field of the (see “RBUF Command Register (rbuf_command, Offset 0x200000)” on page 579) has been set and self cleared. If two firmware entities need to check out buffers through this interface, they must provide their own locking mechanism. Since this interface does the chaining internally, all buffers for a single cluster must be allocated before the firmware lock is released.	RO	0	X	X	X	X
6–0	RESERVED		RO	0	X	X	X	X

RBUF FIRMWARE BUFFER FREE REGISTER (RBUF_FW_BUF_FREE, OFFSET 0x200014)

This register allows buffers to be freed from any processor in the system. It is mandatory that back-to-back writes to this register work properly and all buffers are freed.

Table 419: RBUF Firmware Buffer Free Register (rbuf_fw_buf_free, Offset 0x200014)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–25	RESERVED		RO	0	X	X	–	–
31	FREE_REQ	Setting this bit causes RBUF to free a cluster which is selected by programming its address in the HEAD field.	SC	0	–	–	X	X
30–26	RESERVED		RO	0	–	–	X	X
25	TYPE	This bit determines the type of MBUF that is being freed when the FREE_REQ bit is set. When this bit is set a catchup MBUF will be freed. When this bit is clear an RX traffic MBUF will be freed. This bit is only valid when the CU_ISOLATE bit of the “RBUF Command Register (rbuf_command, Offset 0x200000)” on page 579 is set.			–	–	X	X
24–16	HEAD	This is the head pointer of the MBUF cluster that is to be freed.	RW	0	X	X	X	X
15–7	TAIL	This is the tail pointer of the MBUF cluster that is to be freed.	RW	0	X	X	X	X
6–0	COUNT	This is the number of MBUFs in the cluster to be freed.	RW	0	X	X	X	X

RBUF FIRMWARE BUFFER SELECT REGISTER (RBUF_FW_BUF_SEL, OFFSET 0x200018)

This register allows one cluster of buffers to be selected and accessed through the register interface. Obviously, only one entity may use this interface at any one time. After writing the mbuf cluster value to this register, the data is accessed through the [“RBUF Packet Data Registers \(rbuf_pkt_data\[2250\], Offset 0x208000\)” on page 585](#). The Offset into that area determines the Offset into the cluster.

Table 420: RBUF Firmware Buffer Select Register (rbuf_fw_buf_sel, Offset 0x200018)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–25	RESERVED		RO	0	X	X	–	–
31	SEL_REQ	When this bit is set the RBUF will select an MBUF cluster with the address programmed in the HEAD field.	SC	0	–	–	X	X
30–25	RESERVED		RO	0	–	–	X	X
24–16	HEAD	This is the head pointer of the MBUF cluster that is to be selected.	RW	0	X	X	X	X
15–7	TAIL	This is the tail pointer of the MBUF cluster that is to be selected.	RW	0	X	X	X	X
6–0	COUNT	This is the number of MBUFs in the cluster to be selected.	RW	0	X	X	X	X

RBUF CONFIGURATION 2 REGISTER (RBUF_CONFIG2, OFFSET 0x20001c)*Table 421: RBUF Configuration 2 Register (rbuf_config2, Offset 0x20001c)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–16	MAC_KEEP_TRIP	This is the free count where the RPM will be told to keep packets again. The MAC drop output will be deasserted when the FREE_COUNT field of the RBUF Packet Data Registers (see “ RBUF Packet Data Registers (rbuf_pkt_data[2250], Offset 0x208000) ” on page 585) is greater than or equal to this count.	RW	0x1e	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X
9–0	MAC_DROP_TRIP	This is the free count where the RPM will be told to drop packets. The drop feature is to provide hysteresis and control interactions with the Catchup path. The MAC drop output will be asserted when the FREE_COUNT field of the RBUF Packet Data Registers (see “ RBUF Packet Data Registers (rbuf_pkt_data[2250], Offset 0x208000) ” on page 585) is less than or equal to the level specified in this field. It will stay set until the FREE_COUNT exceeds the value in the MAC_KEEP_TRIP field.	RW	0x5	X	X	X	X

RBUF CONFIGURATION 3 REGISTER (RBUF_CONFIG3, OFFSET 0x200020)*Table 422: RBUF Configuration 3 Register (rbuf_config3, Offset 0x200020)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–26	RESERVED		RO	0	X	X	X	X
25–16	CU_KEEP_TRIP	This is the free count where the Catchup Parser will be told to keep packets again. The Catchup Parser drop output will be deasserted when the FREE_COUNT field of the RBUF Packet Data Registers (see “ RBUF Packet Data Registers (rbuf_pkt_data[2250], Offset 0x208000) ” on page 585) is greater than or equal to this count.	RW	0x1e	X	X	X	X
15–10	RESERVED		RO	0	X	X	X	X

Table 422: RBUF Configuration 3 Register (rbuf_config3, Offset 0x200020) (Cont.)

Bit	Name	Description	Mode	Reset	06	08	09	16
9-0	CU_DROP_TRIP	This is the free count where the Catchup Parser will be told to drop packets. The drop feature is to provide hysteresis and control interactions with the Catchup path. The Catchup Parser drop output will be asserted when the FREE_COUNT field of the RBUF Packet Data Registers (see "RBUF Packet Data Registers (rbuf_pkt_data[2250], Offset 0x208000)" on page 585) is less than or equal to the level specified in this field. It will stay set until the FREE_COUNT field exceeds the value in the CU_KEEP_TRIP field.	RW	0x12	X	X	X	X

RBUF PACKET DATA REGISTERS (RBUF_PKT_DATA[2250], OFFSET 0x208000)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

RBUF CLUSTER LIST DATA REGISTERS (RBUF_CLIST_DATA[512], OFFSET 0x210000)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

RBUF MBUF DATA REGISTERS (RBUF_BUF_DATA[16384], OFFSET 0x220000)

Description	Mode	Reset	06	08	09	16
	RW	0	X	X	X	X

PCI EXPRESS TO GRC INTERFACE (P2R) BLOCK

P2R EPB CONFIGURATION REGISTERS (P2R_EPB_CONFIG[256], OFFSET 0x240000)

Description	Mode	Reset	06	08	09	16
These registers provide access to the EPB PCIe configuration space. All accesses to the PCIe configuration space are 32-bit aligned accesses through this area.	RW	0	-	X	-	-

P2R DEBUG REGISTERS (P2R_DEBUG[256], OFFSET 0x240400)

Description	Mode	Reset	06	08	09	16
These read/write registers provide access to the EPB PCIe debug register space.	RW	0	-	X	-	-

P2R MDIO ADDRESS REGISTER (P2R_MDIO_ADDR, OFFSET 0x240800)

Table 423: P2R MDIO Address Register (p2r_mdio_addr, Offset 0x240800)

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	-	X	X
31–20	CMD	Writing a 0 to this field causes no action and allows the address to be programmed in preparation for a write. Writing a 1 to this field initiates a read at the address and port specified.	RW	0	-	X	-	-
19–16	PORT	This value controls the port address sent on the MDIO bus connected to the PCIe SerDes block.	RW	0	-	X	-	-
15–0	ADR	This value controls the register index sent on the MDIO bus connected to the PCIe SerDes block.	RW	0	-	X	-	-

P2R MDIO WRITE DATA REGISTER (P2R_MDIO_WR_DATA, OFFSET 0x240804)*Table 424: P2R MDIO Write Data Register (p2r_mdio_wr_data, Offset 0x240804)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	–	X	X
31	CMD	This bit must be set to initiate a write cycle based on the DATA field and the ADR field of the “ P2R MDIO Address Register (p2r_mdio_addr, Offset 0x240800) ” on page 586. When the write completes, this bit is cleared.	SC	0	–	X	–	–
30–16	RESERVED		RO	0	–	X	–	–
15–0	DATA	This value is the register data for write cycles.	RW	0	–	X	–	–

P2R MDIO READ DATA REGISTER (P2R_MDIO_RD_DATA, OFFSET 0x240808)*Table 425: P2R MDIO Read Data Register (p2r_mdio_rd_data, Offset 0x240808)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–0	RESERVED		RO	0	X	–	X	X
31	CMD	This bit reads as 0 until a requested read of the PCIe SerDes has completed and then it reads as 1. This bit is automatically cleared by a write to the “ P2R MDIO Address Register (p2r_mdio_addr, Offset 0x240800) ” on page 586.	RO	0	–	X	–	–
30–16	RESERVED		RO	0	–	X	–	–
15–0	DATA	After a read has been requested in the ADR field of the “ P2R MDIO Address Register (p2r_mdio_addr, Offset 0x240800) ” on page 586, this area returns the MDIO data. This field is only valid if the CMD bit is set.	RO	0	–	X	–	–

P2R COMMAND REGISTER (P2R_COMMAND, OFFSET 0x244000)*Table 426: P2R Command Register (p2r_command, Offset 0x244000)*

Bit	Name	Description	Mode	Reset	06	08	09	16
31–1	RESERVED		RO	0	X	X	X	X
0	RESERVED		RO	0	X	–	X	X
0	P2R_CMD_GRC_TIMEOUT	Setting this bit enables a 256-ms timeout. If the access is not completed by the EPB block within the timeout time, the value 0xDEADBEEF is returned. If this bit is clear, then accesses to the EPB waits indefinitely.	RW	0	–	X	–	–

Section 11: EPB Register Definitions



Note: For BCM5709 only.

The NetXtreme II family of controllers is available with interfaces for both the PCI-X and the PCI Express® bus. On PCI Express-based BCM5708, the Ethernet controller device is implemented behind a built-in PCI Express bridge device which connects to the Ethernet controller device through a Private Interconnect Bus (or PIB). This PCI Express bridge device, known as the EPB, is normally configured through standard PCI configuration mechanisms in both the system BIOS and the operating system, and is compliant with the PCI Express 1.0a and PCI Express to PCI-X Bridge specifications.

EPB FEATURES

The EPB has the following major features:

- Full support for the PCI Express protocol, up to a maximum x4 link width
- ASPM L0s and L1 support
- L1 and L2 (low power) states
- Beacon and WAKE# support
- PME Message support
- Support of Legacy PCI interrupt messaging
- PCI Express port is identified as an upstream port of a downstream device.
- PCI Express Advanced Error Reporting
- RAS Features

EPB QUEUES/BUFFERS

The EPB provides the following buffers and queues:

- Four-deep downstream memory write request queue
- Four-deep downstream non posted request queue
- Four-deep upstream write request queue.
- Eight-deep upstream read request queue.
- Four-deep 32-byte buffers for downstream posted writes
- Four-deep 4-byte buffer for downstream non-posted writes.
- One-deep 64-byte buffer for downstream Reads
- 1 K Byte buffer for buffering upstream write data
- 1 K Byte buffer for buffering upstream read data (read responses)

EPB CONFIGURATION REGISTERS

The EPB configuration registers map into the PCI Express configuration space. The EPB interfaces to the PCI Express port of the upstream component through the PCI Express link and responds to any Type-0 configuration cycle. The EPB supports one function: Function 0. The upstream component forwards configuration cycles targeted to the EPB as Type-0 PCI configuration cycles on the PCI Express interface. This section describes the register interface for the PCI Express Interface Module. The PCI Express interface is referred to as the primary interface and the PIB is referred to as the secondary interface.

VENDOR ID REGISTER (EPB_VID, OFFSET 0x00)



Note: For BCM5708 only.

The read-only Vendor ID register identifies the manufacturer of the PCI adapter and defaults to 0x1166 at power-on reset. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness

Table 427: Vendor ID Register (epb_vid, Offset 0x00)

Bit	Field	Description	Mode	Reset	08
15–0	VID	Unique identifier for the PCI adapter manufacturer.	RO	0x1166	X

DEVICE ID REGISTER (EPB_DID, OFFSET 0x02)



Note: For BCM5708 only.

The read-only Device ID register identifies a particular adapter within those made by the same manufacturer and defaults to 0x0103 at power-on reset

Table 428: Device ID Register (epb_did, Offset 0x02)

Bit	Field	Description	Mode	Reset	08
15–0	DID	Unique identifier for the EPB bridge.	RO	0x0103	X

COMMAND REGISTER (EPB_CR, OFFSET 0x04)**Note:** For BCM5708 only.

The read/write Command register is used to enable various features of the device. All of the bit positions are predefined by the PCI specification. Not all bits defined in the PCI specification are implemented by this register.

Table 429: Command Register (epb_cr, Offset 0x04)

Bit	Field	Description	Mode	Reset	08
15–9	RESERVED		RO	0x0	X
8	SERR	When this bit is set it enables reporting of non-fatal and fatal errors to the root complex. In addition, setting this bit enables transmission by the PCIE interface of ERR_NONFATAL and ERR_FATAL error messages on behalf of SERR assertions detected on the PIB interface. Errors are reported when enabled either through this bit or through the PCIe specific bits in the Device Control register.	RW	0	X
7	RESERVED		RO	0	X
6	PAR_ERR_RESP	Setting this bit enables the EPB setting of the Master Data Parity Error bit in the Status register in response to receiving a corrupted TLP on the PCIe interface. This bit does not control the forwarding of corrupted data from PCIe to PIB; i.e., a received corrupted TLP is forwarded with bad Parity to PIB regardless of the setting of this bit.	RW	0	X
5–3	RESERVED	RESERVED	RO	0	X
2	BMSTR	Setting this bit enables the EPB to operate as a master on the PCIe interface for memory requests forwarded from the PIB interface.	RW	0	X
1	MEM	Setting this bit enables forwarding of memory transactions to the PIB interface and any internal functions. Clearing this bit will cause the EPB to respond to all memory requests on the PCIe interface as Unsupported Request Completion. All memory requests from the PIB interface will be forwarded to the PCIe interface.	RW	0	X
0	IO	Setting this bit will enable the EPB to forward I/O requests to the PIB interface. Clearing this bit will cause the EPB to respond to all I/O requests on the PCIe interface with an unsupported request completion.	RO	0	X

STATUS REGISTER (EPB_SR, OFFSET 0x06)**Note:** For BCM5708 only.**Table 430: Status Register (epb_sr, Offset 0x06)**

Bit	Name	Description	Mode	Reset	08
15	PAR_ERR	When this bit is set it indicates that the EPB has received a corrupted TLP (read completion or write request) on the PCIe interface, regardless of the state of the PERR_ENA bit of the "Command Register (epb_cr, Offset 0x04)" on page 591.	WC	0	X
14	SIG_SYS_ERR	When this bit is set it indicates that the EPB has signaled an ERR_FATAL or ERR_NONFATAL message to the Root complex and the SERR_ENA bit of the "Command Register (epb_cr, Offset 0x04)" on page 591 device is set.	WC	0	X
13	RCV_MSTR_ABO RT	When this bit is set it indicates that the EPB has received a Completion with unsupported request completion status on the PCIe interface.	WC	0	X
12	RCV_TGT_ABOR T	When this bit is set it indicates that the EPB has received a completion with completer abort completion status on the PCIe interface.	WC	0	X
11	SIG_TGT_ABOR T	When this bit is set it indicates that the EPB has generated a completion with completer abort completion status in response to a request received on the PCIe interface.	WC	0	X
10–9	RESERVED	RESERVED	RO	0	X
8	MSTR_PAR_ERR	When this bit is set it indicates that the EPB has detected an uncorrectable data error on the PCIe interface and the PERR_ENA bit of the "Command Register (epb_cr, Offset 0x04)" on page 591 device is set. This condition may occur when the EPB receives a completion with data marked corrupted or when the EPB poisons a write request on the PCIe interface.	WC	0	X
7–5	RESERVED	RESERVED	RO	0x0	X
4	PIB_CAP_LIST_I TEM	This bit is tied high to indicate that the EPB supports a capabilities list.	RO	0x1	X
3–0	RESERVED	RESERVED	RO	0	X

REVISION ID REGISTER (EPB_REVISION_ID, OFFSET 0x08)



Note: For BCM5708 only.

This read-only register identifies the revision of the EPB. The power-on reset value varies depending on the BCM5708 revision in use.

CLASS CODE REGISTER (EPB_CLASS_CODE, OFFSET 0x09)



Note: For BCM5708 only.

This read-only Class Code register identifies the generic function of the device and has the value of 0x060400 at power-on reset.

HEADER TYPE REGISTER (EPB_HDR_TYPE, OFFSET 0x0E)



Note: For BCM5708 only.

The read-only Header Type register identifies both the layout of bytes 0x10 through 0x3f of the configuration space as well as whether the adapter supports multiple PCI functions. This register is always 0x01 which indicates that this is a single function PCI-to-PCI bridge device.

Table 431: Header Type Register (epb_hdr_type, Offset 0x0e)

Bit	Field	Description	Mode	Reset	08
7–0	HEADER_TYPE	Type 0x01 configuration space.	RO	0x01	X

PRIMARY BUS NUMBER (EPB_BSNUM, OFFSET 0x18)



Note: For BCM5708 only.

This register contains the primary bus number (or the upstream bus) that is attached to the PCIe interface of the EPB.

Table 432: Primary Bus Number (epb_bsnum, Offset 0x18)

Bit	Field	Description	Mode	Reset	08
7–0	BSNUM	Primary bus number.	RW	0x00	X

SECONDARY PRIMARY BUS NUMBER (EPB_SBSNUM, OFFSET 0x19)



Note: For BCM5708 only.

This register contains the secondary bus number (or the downstream bus) that is attached to the PIB interface of the EPB.

Table 433: Secondary Primary Bus Number (epb_sbsnum, Offset 0x19)

Bit	Field	Description	Mode	Reset	08
7–0	SBSNUM	Secondary bus number.	RW	0x00	X

SUBORDINATE BUS NUMBER (EPB_SUBBSNUM, OFFSET 0x1A)



Note: For BCM5708 only.

This register contains the highest numbered PCI bus which is downstream of (or subordinate to) the EPB.

Table 434: Subordinate Bus Number (epb_subbsnum, Offset 0x1a)

Bit	Field	Description	Mode	Reset	08
7–0	SUBBSNUM	Secondary bus number.	RW	0x00	X

SECONDARY LATENCY TIMER (EPB_SLTR, OFFSET 0x1B)



Note: For BCM5708 only.

This register defines the timeslice when the EPB is acting as the initiator on the PIB.

Table 435: Secondary Latency Timer (epb_sltr, Offset 0x1b)

Bit	Field	Description	Mode	Reset	08
7–3	VALUE	This field controls the latency timer for the PIB.	RW	0x00	X

Table 435: Secondary Latency Timer (epb_sltr, Offset 0x1b) (Cont.)

Bit	Field	Description	Mode	Reset	08
2–0	RESERVED	RESERVED	RO	0	X

I/O BASE REGISTER (EPB_IOBR, OFFSET 0x1c)**Note:** For BCM5708 only.

This register defines the I/O base address for the I/O memory decoded behind the EPB bridge.

Table 436: I/O Base Register (epb_iobr, Offset 0x1c)

Bit	Field	Description	Mode	Reset	08
7–4	VALUE	This field corresponds to the upper four bits (address bits 15–12) of the PCIe I/O address request.	RW	0x0	X
3–0	RESERVED	These bits are hardwired to 0 indicating 16-bit I/O support.	RO	0	X

I/O LIMIT REGISTER (EPB_IOLR, OFFSET 0x1d)**Note:** For BCM5708 only.

This register defines the limit for the I/O memory decoded behind the EPB bridge. The EPB responds to all I/O accesses that satisfy the following condition:

$$\text{IO_BASE_ADDR} \leq \text{I/O Address [15:12]} \leq \text{IO_LIMIT}$$

Table 437: I/O Limit Register (epb_iolr, Offset 0x1d)

Bit	Field	Description	Mode	Reset	08
7–4	VALUE	This field corresponds to the upper four bits (address bits 15–12) of the PCIe I/O address request.	RW	0x0	X
3–0	RESERVED	These bits are hardwired to 0 indicating 16-bit I/O support.	RO	0	X

SECONDARY STATUS REGISTER (EPB_SSR, OFFSET 0x1E)**Note:** For BCM5708 only.**Table 438: Secondary Status Register (epb_ssr, Offset 0x1e)**

Bit	Name	Description	Mode	Reset	08
15	DET_PAR_ERR	This bit is set when the EPB detects an uncorrectable address, attribute, or data error on the PIB interface. This bit will be set when any of the following conditions are true, regardless of the state of the PERR_ENA bit of the “Bridge Control Register (epb_bridge_cr, Offset 0x3e)” on page 599: <ul style="list-style-type: none"> The EPB detects an uncorrectable address or attribute error as a potential target The EPB detects an uncorrectable data error when it is the target of a write transaction or PIB split completion The EPB detects an uncorrectable data error when it is the master of a read transaction (immediate read data or PIB split response). 	WC	0	X
14	RCV_SYS_ERR	This bit reports the detection of an SERR# assertion on the PIB interface of the EPB.	WC	0	X
13	RCV_MSTR_ABR_T_ERR	This bit reports the detection of a master abort termination by the EPB when it is the master of a transaction on the PIB interface or when receiving a PIB split completion message indicating master abort.	WC	0	X
12	RCV_TGT_ABRT_ERR	This bit reports the detection of a target abort termination by the EPB when it is the master of a transaction on the PIB interface or when receiving a PIB split completion message indicating target abort.	WC	0	X
11	SIG_TGT_ABRT_ERR	This bit reports the signaling of a target abort termination by the EPB when it responds as the target of a transaction on its PIB interface or when it signals a PIB split completion with target abort.	WC	0	X
10–9	DEVSEL_TMG	These bits encode the slowest timing of DEVSEL, except for configuration cycles. Value entries are 00 for fast, 01 for medium and 10 for slow. The device is hardwired for medium DEVSEL response speed.	RO	01	X
8	MSTR_DATA_PA_R_ERR	This bit reports the detection of an uncorrectable data error by the EPB. This bit is set when the EPB is the bus master of the transaction on the PIB interface, the PERR_ENA bit of the “Bridge Control Register (epb_bridge_cr, Offset 0x3e)” on page 599 is set, and any of the following occur: <ul style="list-style-type: none"> The EPB asserts PERR# on a read transaction The EPB detects PERR# asserted on a write transaction The EPB receives a split completion message for a non-posted write indicating an uncorrectable (split) write data error The EPB receives an uncorrectable data error in a split completion or split completion message 	WC	0	X
7–6	RESERVED	RESERVED	RO	0x0	X

Table 438: Secondary Status Register (epb_ssr, Offset 0x1e) (Cont.)

Bit	Name	Description	Mode	Reset	08
5	66_CAP	When this bit is set it indicates that the PIB interface is capable of operating at 66 MHz in conventional PCI mode.	RO	0	X
4–0	RESERVED	RESERVED	RO	0	X

MEMORY BASE REGISTER (EPB_MBR, OFFSET 0x20)**Note:** For BCM5708 only.

This register defines the memory base address for the memory decoded behind the EPB bridge.

Table 439: Memory Base Register (epb_mbr, Offset 0x20)

Bit	Field	Description	Mode	Reset	08
15–4	VALUE	This field corresponds to the upper 12 bits (address bits 31–20) of the PCIe memory address request.	RW	0x0	X
3–0	RESERVED	These bits are hardwired to 0.	RO	0	X

MEMORY LIMIT REGISTER (EPB_MLR, OFFSET 0x22)**Note:** For BCM5708 only.

This register defines the limit for the memory decoded behind the EPB bridge. The EPB responds to all memory accesses that satisfy the following condition:

$$\text{mbr}[\text{VALUE}] \leq \text{Memory Address [31-20]} \leq \text{mlr}[\text{VALUE}]$$

Table 440: Memory Limit Register (epb_mlr, Offset 0x22)

Bit	Field	Description	Mode	Reset	08
15–4	VALUE	This field corresponds to the upper 12 bits (address bits 31–20) of the PCIe memory address request.	RW	0x0	X
3–0	RESERVED	These bits are hardwired to 0.	RO	0	X

PREFETCHABLE MEMORY BASE REGISTER (EPB_PMBR, OFFSET 0x24)



Note: For BCM5708 only.

Table 441: Prefetchable Memory Base Register (epb_pmbr, Offset 0x24)

Bit	Field	Description	Mode	Reset	08
15–4	VALUE	This field corresponds to the upper 12 bits (address bits 31–20) of the PCIe memory address request.	RW	0x0	X
3–0	RESERVED	These bits are hardwired to 0x1 to indicate 64-bit addressing support.	RO	0x1	X

PREFETCHABLE MEMORY LIMIT REGISTER (EPB_PMLR, OFFSET 0x26)



Note: For BCM5708 only.

Table 442: Prefetchable Memory Limit Register (epb_pmlr, Offset 0x26)

Bit	Field	Description	Mode	Reset	08
15–4	VALUE	This field corresponds to the upper 12 bits (address bits 31–20) of the PCIe memory address request.	RW	0x0	X
3–0	RESERVED	These bits are hardwired to 0x1 to indicate 64-bit addressing support.	RO	0x1	X

PREFETCHABLE MEMORY BASE UPPER 32BIT REGISTER (EPB_PMUBR, OFFSET 0x28)



Note: For BCM5708 only.

Table 443: Prefetchable Memory Base Upper 32bit Register (epb_pmubr, Offset 0x28)

Bit	Field	Description	Mode	Reset	08
31–0	VALUE	This field corresponds to the upper 32 bits (address bits 63–31) of the PCIe memory address request.	RW	0x0	X

PREFETCHABLE MEMORY LIMIT UPPER 32-BIT REGISTER (EPB_PMULR, OFFSET 0x2c)**Note:** For BCM5708 only.**Table 444: Prefetchable Memory Limit Upper 32-bit Register (epb_pmulr, Offset 0x2c)**

Bit	Field	Description	Mode	Reset	08
31–0	VALUE	This field corresponds to the upper 32 bits (address bits 63–32) of the PCIe memory address request.	RW	0x0	X

CAPABILITIES POINTER REGISTER (EPB_NEXT_CAP_34, OFFSET 0x34)**Note:** For BCM5708 only.

The read-only Capabilities Pointer register specifies an offset in the PCI address space of a linked list of new capabilities and has the value 0x60 at power-on reset. The PCI Express and PCI-X capabilities registers are supported.

BRIDGE CONTROL REGISTER (EPB_BRIDGE_CR, OFFSET 0x3E)**Note:** For BCM5708 only.**Table 445: Bridge Control Register (epb_bridge_cr, Offset 0x3e)**

Bit	Name	Description	Mode	Reset	08
15–7	RESERVED	RESERVED	RO	0	X
6	SEC_RST	Setting this bit forces assertion of RST on the PIB interface. The EPB's PIB interface and any buffers between the two interfaces are initialized to the default state whenever this bit is set. The PCIe interface and all configuration space registers are not affected by setting this bit. Because PIB RST# is asserted for as long as this bit is set, software must observe proper PCI-X reset timing requirements.	RW	1	X
5	MASTER_ABORT	Master abort mode	RW	1	X
5–2	RESERVED	RESERVED	RO	0	X

Table 445: Bridge Control Register (epb_bridge_cr, Offset 0x3e) (Cont.)

Bit	Name	Description	Mode	Reset	08
1	SERR_ENA	The EPB transmits an ERR_FATAL or ERR_NONFATAL cycle on the PCIe interface when all of the following are true: <ul style="list-style-type: none">• SERR# is asserted on the PIB interface• This bit is set or the SERR# Assertion Detected Mask bit is clear in the Secondary Uncorrectable Error Mask register• The SERR# Enable bit is set in the Command register or the PCIe-specific bits are set in the Device Control register of the PCIe capability structure			X
0	PERR_RESP_ENA	When this bit is set it enables uncorrectable address, attribute, and data error detection and reporting on the PIB interface.	RW	0	X

PCI EXPRESS CAPABILITIES REGISTER (EPB_PCIE_CAP_ID, OFFSET 0x60)

Note: For BCM5708 only.

The read-only PCI Express Capability ID register is set to 0x10 to indicate that the next 20 bytes contain a PCI Express capabilities block.

PCI EXPRESS NEXT CAPABILITY POINTER REGISTER (EPB_PCIE_NEXT_CAP_PTR, OFFSET 0x61)



Note: For BCM5708 only.

The read-only Next Capability register continues the PCI capability chain and has the value 0x90 at power-on reset. Its value specifies an offset in the PCI address space of the next capability.

PCI EXPRESS CAPABILITIES REGISTER (EPB_PCIE_CAPABILITY, OFFSET 0x62)



Note: For BCM5708 only.

Table 446: PCI Express Capabilities Register (epb_PCIE_capability, Offset 0x62)

Bit	Field	Description	Mode	Reset	08
15–8	RESERVED		RO	0x0	X
7–4	DEVICE_PORT_TYPE	This field indicates that the device is an Express to PCI-X bridge.	RO	0x7	X
3–0	CAP_VERSION	This field indicates the PCI Express capability structure version number.	RO	0x1	X

PCI EXPRESS DEVICE CAPABILITIES REGISTER (EPB_PCIE_DEV_CAPABILITY, OFFSET 0x64)



Note: For BCM5708 only.

Table 447: PCI Express Device Capabilities Register (epb_PCIE_dev_capability, Offset 0x64)

Bit	Name	Description	Mode	Reset	08
31–12	RESERVED		RO	0x0	X
11–9	L1_LATENCY	This field indicates the acceptable latency that an endpoint can withstand due to a transition from the L1 state to the L0 state.	RO	0x4	X
Value	Name	Description			
0	1US	Less than 1 µs			
1	2US	1 µs to less than 2 µs			
2	4US	2 µs to less than 4 µs			
3	8US	4 µs to less than 8 µs			
4	16US	8 µs to less than 16 µs			
5	32US	16 µs to less than 32 µs			
6	64US	32 µs to 64 µs			
7	LARGE	Greater than 64 µs			

Table 447: PCI Express Device Capabilities Register (epb_pcic_dev_capability, Offset 0x64) (Cont.)

Bit	Name	Description	Mode	Reset	08																											
8–6	L0S_LATENCY	This field indicates the acceptable latency that an endpoint can withstand due to a transition from the L0S state to the L0 state.	RO	0x6	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr><td>0</td><td>64US</td><td>Less than 64 µs</td></tr> <tr><td>1</td><td>128US</td><td>64 µs to less than 128 µs</td></tr> <tr><td>2</td><td>256US</td><td>128 µs to less than 256 µs</td></tr> <tr><td>3</td><td>512US</td><td>256 µs to less than 512 µs</td></tr> <tr><td>4</td><td>1US</td><td>512 µs to less than 1 µs</td></tr> <tr><td>5</td><td>2US</td><td>1 µs to less than 2 µs</td></tr> <tr><td>6</td><td>4US</td><td>2 µs to 4 µs</td></tr> <tr><td>7</td><td>LARGE</td><td>Greater than 4 µs</td></tr> </tbody> </table>	Value	Name	Description	0	64US	Less than 64 µs	1	128US	64 µs to less than 128 µs	2	256US	128 µs to less than 256 µs	3	512US	256 µs to less than 512 µs	4	1US	512 µs to less than 1 µs	5	2US	1 µs to less than 2 µs	6	4US	2 µs to 4 µs	7	LARGE	Greater than 4 µs			
Value	Name	Description																														
0	64US	Less than 64 µs																														
1	128US	64 µs to less than 128 µs																														
2	256US	128 µs to less than 256 µs																														
3	512US	256 µs to less than 512 µs																														
4	1US	512 µs to less than 1 µs																														
5	2US	1 µs to less than 2 µs																														
6	4US	2 µs to 4 µs																														
7	LARGE	Greater than 4 µs																														
5	EXT_TAG_FIELD	This field indicates the maximum supported size of the tag field when this PCI function acts as a requester.	RO	1	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr><td>0</td><td>FIVE_BIT_TAG</td><td>5-bit tag field supported</td></tr> <tr><td>1</td><td>EIGHT_BIT_TAG</td><td>8-bit tag field supported</td></tr> </tbody> </table>	Value	Name	Description	0	FIVE_BIT_TAG	5-bit tag field supported	1	EIGHT_BIT_TAG	8-bit tag field supported																					
Value	Name	Description																														
0	FIVE_BIT_TAG	5-bit tag field supported																														
1	EIGHT_BIT_TAG	8-bit tag field supported																														
4–3	PHANTOM_FUNCS	This field indicates the maximum supported size of the tag field when this PCI function acts as a requester.	RO	0x0	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr><td>0</td><td>NONE</td><td>No phantom functions.</td></tr> <tr><td>1</td><td>1</td><td>The MSB of the function number in the Requester ID field is used for phantom functions.</td></tr> <tr><td>2</td><td>2</td><td>The two MSBs of the function number in the Requester ID field are used for phantom functions.</td></tr> <tr><td>3</td><td>3</td><td>All three bits of the function number in the Requester ID field are used for phantom functions.</td></tr> </tbody> </table>	Value	Name	Description	0	NONE	No phantom functions.	1	1	The MSB of the function number in the Requester ID field is used for phantom functions.	2	2	The two MSBs of the function number in the Requester ID field are used for phantom functions.	3	3	All three bits of the function number in the Requester ID field are used for phantom functions.															
Value	Name	Description																														
0	NONE	No phantom functions.																														
1	1	The MSB of the function number in the Requester ID field is used for phantom functions.																														
2	2	The two MSBs of the function number in the Requester ID field are used for phantom functions.																														
3	3	All three bits of the function number in the Requester ID field are used for phantom functions.																														
2–0	MAX_PAYLOAD	This field indicates the maximum payload size supported by the device.	RO	0x0	X																											
		<table border="1"> <thead> <tr> <th>Value</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr><td>0</td><td>128B</td><td>128-bytes max payload size</td></tr> <tr><td>1</td><td>256B</td><td>256-bytes max payload size</td></tr> <tr><td>2</td><td>512B</td><td>512-bytes max payload size</td></tr> <tr><td>3</td><td>1KB</td><td>1024-bytes max payload size</td></tr> <tr><td>4</td><td>2KB</td><td>2048-bytes max payload size</td></tr> <tr><td>5</td><td>4KB</td><td>4096-bytes max payload size</td></tr> <tr><td>6</td><td>RESERVED</td><td></td></tr> <tr><td>7</td><td>RESERVED</td><td></td></tr> </tbody> </table>	Value	Name	Description	0	128B	128-bytes max payload size	1	256B	256-bytes max payload size	2	512B	512-bytes max payload size	3	1KB	1024-bytes max payload size	4	2KB	2048-bytes max payload size	5	4KB	4096-bytes max payload size	6	RESERVED		7	RESERVED				
Value	Name	Description																														
0	128B	128-bytes max payload size																														
1	256B	256-bytes max payload size																														
2	512B	512-bytes max payload size																														
3	1KB	1024-bytes max payload size																														
4	2KB	2048-bytes max payload size																														
5	4KB	4096-bytes max payload size																														
6	RESERVED																															
7	RESERVED																															

PCI EXPRESS DEVICE CONTROL REGISTER (EPB_PCIE_CONTROL, OFFSET 0x68)**Note:** For BCM5708 only.**Table 448: PCI Express Device Control Register (epb_PCIE_control, Offset 0x68)**

Bit	Name	Description	Mode	Reset	08
15	RESERVED		RO	0	X
14–12	MAX_READ_REQ	Maximum read request size.	RO	0x0	X
	Value	Name	Description		
	0	128B	128-bytes max payload size		
	1	256B	256-bytes max payload size		
	2	512B	512-bytes max payload size		
	3	1KB	1024-bytes max payload size		
	4	2KB	2048-bytes max payload size		
	5	4KB	4096-bytes max payload size		
	6	RESERVED			
	7	RESERVED			
11	NO_SNOOP	Enable no snoop	RO	1	X
10	AUX_PWR_PM_ENA	Auxiliary power PM enable	RWS	0	X
9	PHANTOM_FUNCS_ENA	When this bit is set phantom functions are enabled.	RO	0	X
8	EXT_TAG_ENA	When this bit is set it enables the device to use an 8-bit tag field as a requester.	RW	0	X
7–5	MAX_PAYLOAD_SIZE	The EPB ignores writes to this field and does not initiate requests greater than 128 bytes.	RW	0x0	X
4	RELAX_ORDER_ENA	When this bit is set the device is permitted to set the Relaxed Ordering bit in the Attributes field of the requests it initiate that do not require strong write ordering.	RO	0	X
3	UNSUPP_REQ_ENA	When this bit is set the device enables reporting of unsupported requests.	RW	0	X
2	FATAL_ERR_ENA	When this bit is set the device enables reporting of fatal errors.	RW	0	X
1	NON_FATAL_ERR_ENA	When this bit is set the device enables reporting of non-fatal errors.	RW	0	X
0	CORR_ERR_ENA	When this bit is set the device enables reporting of correctable errors.	RW	0	X

PCI EXPRESS DEVICE STATUS REGISTER (EPB_PCIE_DEV_STATUS, OFFSET 0x6A)**Note:** For BCM5708 only.**Table 449: PCI Express Device Status Register (epb_PCIE_DEV_STATUS, Offset 0x6A)**

Bit	Field	Description	Mode	Reset	08
31–6	RESERVED		RO	0x0	X
5	TRANS_PEND	When this bit is set it indicates that the device has issued a nonposted request which has not completed.	WC	0	X
4	RESERVED		RO	0	X
3	UNSUPP_REQ_DET	When this bit is set it indicates that the device has received an unsupported request.	WC	0	X
2	FATAL_ERR_DET	When this bit is set it indicates that the device has detected one or more fatal errors since the last time this bit was cleared.	WC	0	X
1	NON_FATAL_ERR_DET	When this bit is set it indicates that the device has detected one or more non-fatal errors since the last time this bit was cleared.	WC	0	X
0	CORR_ERR_DET	When this bit is set it indicates that the device has detected one or more correctable errors since the last time this bit was cleared.	RO	0x1	X

PCI EXPRESS LINK CAPABILITIES REGISTER (EPB_PCIE_LINK_CAPABILITY, OFFSET 0x6C)**Note:** For BCM5708 only.**Table 450: PCI Express Link Capabilities Register (epb_PCIE_LINK_CAPABILITY, Offset 0x6C)**

Bit	Name	Description	Mode	Reset	08
31–24	PORT	This field indicates the PCI Express port number for the given PCI Express link.	RO	0xXX	X
23–18	RESERVED		RO	0x0	X
17–15	L1_EXIT_LAT	L1 exit latency.	RO	0x2	X
Value			Description		
0			Less than 1 μ s		
1			1 μ s to less than 2 μ s		
2			2 μ s to less than 4 μ s		
3			4 μ s to less than 8 μ s		
4			8 μ s to less than 16 μ s		
5			16 μ s to less than 32 μ s		
6			32 μ s to 64 μ s		
7			Greater than 64 μ s		

Table 450: PCI Express Link Capabilities Register (epb_pcie_link_capability, Offset 0x6c) (Cont.)

Bit	Name	Description			Mode	Reset	08
14–12	L0S_EXIT_LAT	L0S exit latency.			RO	0xX	X
		Value	Name	Description			
		0	64NS	Less than 64 µs			
		1	128NS	64 µs to less than 128 µs			
		2	256NS	128 µs to less than 256 µs			
		3	512NS	256 µs to less than 512 µs			
		4	1US	512 µs to less than 1 µs			
		5	2US	1 µs to less than 2 µs			
		6	4US	2 µs to less than 4 µs			
		7	LARGE	Greater than 4 µs			
11–10	ASPM_SUPP	ASPM support.			RO	0x3	X
		Value	Name	Description			
		0	RESERVED				
		1	L0S_SUPP	L0s entry supported			
		2	RESERVED				
		3	L0S_L1_SUPP	L0s and L1 entry supported			
9–4	MAX_LINK_WIDTH	Maximum link width.			RO	0x4	X
		Value	Name	Description			
		0	RESERVED				
		0x01	x1	PCI Express x1 link			
		0x02	x2	PCI Express x2 link			
		0x04	x4	PCI Express x4 link			
		0x08	x8	PCI Express x8 link			
		0x0c	x12	PCI Express x12 link			
		0x20	x16	PCI Express x16 link			
		All others	RESERVED				
3–0	MAX_LINK_SPD	Maximum link speed.			RO	0x1	X
		Value	Name	Description			
		1	2_5G	2.5 Gbps			
		All others	RESERVED				

PCI EXPRESS LINK CONTROL REGISTER (EPB_PCIE_LINK_CONTROL, OFFSET 0x70)



Note: For BCM5708 only.

Table 451: PCI Express Link Control Register (epb_PCIE_link_control, Offset 0x70)

Bit	Name	Description	Mode	Reset	08
15–8	RESERVED		RO	0x0	X
7	EXT_SYNC	When this bit is set it forces the transmission of 4096 FTS ordered sets during L0s state, followed by a single SKP ordered set prior to entering the L0 state, and 1024 TS1 ordered sets in the L1 state prior to entering the Recovery state.	RW	0	X
6	COMMON_CLK	When this bit is set in indicates that this device and the device at the other end of the link are using a common reference clock.	RW	0	X
5–4	RESERVED		RO	0x0	X
3	READ_COMP_BOUNDARY	When this bit is set it indicates that the read control boundary is 128 bytes; otherwise, it is 64 bytes.	RO	0x0	X
2	RESERVED		RO	0	X
1–0	ASPM_CONTROL	ASPM control.	RO	0x1	X
Value	Name	Description			
0	DISABLED	Disabled			
1	L0S_ENTRY_ENA	L0s entry enabled			
2	L1_ENTRY_ENA	L1 entry enabled			
3	L0S_L1_ENTRY_ENA	L0s and L1 entry enabled			

PCI EXPRESS LINK STATUS REGISTER (EPB_PCIE_LINK_STATUS, OFFSET 0x72)**Note:** For BCM5708 only.**Table 452: PCI Express Link Status Register (epb_PCIE_link_status, Offset 0x72)**

Bit	Name	Description	Mode	Reset	08
15–13	RESERVED		RO	0x0	X
12	SLOT_CLK_CFG	When this bit is set it indicates that the device uses the same physical reference clock provided by the platform.	RO	X	X
11	LINK_TRAIN	When this bit is set it indicates that Link training is in progress or that the Retrain Link bit was set but Link training has not begun.	RO	X	X
10	TRAIN_ERR	When this bit is set it indicates that a link training error has occurred.	RO	X	X
9–4	NEG_LINK_WIDTH	Negotiated link width.	RO	0xXX	X
		Value Name Description			
		0x01 x1 PCI Express x1 link			
		0x02 x2 PCI Express x2 link			
		0x04 x4 PCI Express x4 link			
		0x08 x8 PCI Express x8 link			
		0x0c x12 PCI Express x12 link			
		0x10 x16 PCI Express x16 link			
	All others	RESERVED			
3–0	NEG_LINK_SPEED	Negotiated link speed.	RO	0xX	X
		Value Name Description			
		1 2_5G 2.5 Gbps			
	All others	RESERVED			

PCI EXPRESS RECEIVE RESOURCE REGISTER (EPB_PCIE_RCV_RESP, OFFSET 0x74)**Note:** For BCM5708 only.**Table 453: PCI Express Receive Resource Register (epb_PCIE_rcv_resp, Offset 0x74)**

Bit	Field	Description	Mode	Reset	08
15–12	RESERVED		RO	0x0	X
11–8	POST_WRT_CREDITS	Posted write credits	RW	0x8	X
7–4	NON_POST_HDR_CREDITS	Non-posted header credits	RW	0x4	X
3–0	POST_HDR_CREDITS	Posted header credits	RO	0x4	X

PCI EXPRESS LINK SPECIAL ERROR STATUS REGISTER (EPB_PCIE_LINK_SPECIAL_ERR_STATUS, OFFSET 0x76)



Note: For BCM5708 only.

Table 454: PCI Express Link Special Error Status Register (epb_PCIE_LINK_SPECIAL_ERR_STATUS, Offset 0x76)

Bit	Field	Description	Mode	Reset	08
15–7	RESERVED		RO	0x00	X
6	INV_SYM_ERR_LB	When this bit is set it indicates that an invalid symbol has been received in loopback mode.	WC	0	X
5	DATA_CMP_ERR_LB	When this bit is set it indicates that a data compare error has occurred in loopback mode.	WC	0	X
4	ELASTIC_BUF_ERR	When this bit is set it indicates that an error has occurred in the elastic buffer.	WC	0	X
3	REPLAY_BUF_OVERRUN_ERR	When this bit is set it indicates that a replay buffer overrun error has occurred.	WC	0	X
2	DLP_SYNC_ERR	When this bit is set it indicates that a DLP sync error has occurred.	WC	0	X
1	DLL_SYNC_ERR	When this bit is set it indicates that a DLL sync error has occurred.	—	—	X
0	TLP_SYNC_ERR	When this bit is set it indicates that a TLP sync error has occurred.	WC	0	X

PRIVATE INTERCONNECT BUS CAPABILITY ID REGISTER (EPB_PIB_CAP_ID, OFFSET 0x90)



Note: For BCM5708 only.

The read-only Private Interconnect Bus Capability ID register is set to 0x07 to indicate that the next 8 bytes contain a PCI-X compatible PIB capabilities block.

PRIVATE INTERCONNECT BUS NEXT CAPABILITY POINTER REGISTER (EPB_PIB_NEXT_CAP_PTR, OFFSET 0x91)



Note: For BCM5708 only.

The read-only Next Capability register continues the PCI capability chain and has the value 0xb0 at power-on reset. Its value specifies an offset in the PCI address space of the next capability.

PIB SECONDARY STATUS REGISTER (EPB_PIB_SEC_STATUS, OFFSET 0x92)**Note:** For BCM5708 only.**Table 455: PIB Secondary Status Register (epb_pib_sec_status, Offset 0x92)**

Bit	Name	Description	Mode	Reset	08
15–14	RESERVED				RO 0 X
13–12	PIB_LIST_VER	PIB capabilities list version			RO 0 X
11–10	RESERVED				RO 0 X
9–6	SEC_CLK_FREQ	Secondary clock frequency.	RO	0x3	X
			Value	Name	Description
			0	PCI	PCI mode
			1	PCIX_66	PCI-X 66 Mhz
			2	PCIX_100	PCI-X 100 Mhz
			3	PCIX_133	PCI-X 133 Mhz
			All others	RESERVED	
5	SPLIT_REQ_DELAYED	When this bit is set it indicates that the EPB's secondary interface was forced to delay a split completion transaction forwarded from the EPB's primary interface.	RO	0	X
4	SPLIT_COMP_OVERRUN	When this bit is set it indicates that the EPB's secondary interface issued a retry or disconnect in response to a split completion transaction.	RO	0	X
3	UNEXPECTED_SPLIT_COMP	When this bit is set it indicates that an unexpected split completion is targeted at the EPB from the PIB.	WC	0	X
2	CPLIT_COMP_DISCARD	When this bit is set it indicates that the EPB has discarded a split completion on the PIB because the requestor would not accept it.	WC	0	X
1	133MHZ_CAP	This bit is hardwired to 1 to indicate that the PIB interface is capable of 133-Mhz operation.	RO	1	X
0	64BIT_DEVICE	This bit is hardwired to 1 to indicate that the PIB interface is 64-bits wide.	RO	1	X

PIB BRIDGE STATUS REGISTER (EPB_PIB_PRI_STATUS, OFFSET 0x94)



Note: For BCM5708 only.

Table 456: PIB Bridge Status Register (epb_pib_pri_status, Offset 0x94)

Bit	Field	Description	Mode	Reset	08
31–21	RESERVED		RO	0x000	X
20	SPLIT_COM_OVERRUN	When this bit is set it indicates that the EPB's secondary interface issued a Retry or Disconnect in response to a Split completion transaction.	RO	0	X
19	UNEXPECTED_SPLIT_COMP	When this bit is set it indicates that an unexpected Split completion is targeted at the EPB from the PCIe interface.	RO	0	X
18–16	RESERVED		RO	0	X
15–8	BUS_NUMBER	When read this field returns the EPB's primary bus number.	RO	0x00	X
7–3	DEVICE_NUMBER	When read this field returns the EPB's device number.	RO	0x0	X
2–0	FUNCTION_NUMBER	When read this field returns the EPB's function number.	RO	0x0	X

PCI POWER MANAGEMENT CAPABILITY ID REGISTER (EPB_PM_CAP_ID, OFFSET 0XB0)



Note: For BCM5708 only.

The read-only PCI Express Power Management Capability ID register is set to 0x01 to indicate that the next 6 bytes contain a PCI power management capabilities block.

PCI POWER MANAGEMENT NEXT CAPABILITY POINTER REGISTER (EPB_PM_NEXT_CAP_PTR, OFFSET 0XB1)



Note: For BCM5708 only.

The read-only Next Capability register continues the PCI capability chain and has the value 0x00 at power-on reset, indicating that this is the last capabilities block in the chain.

PCI POWER MANAGEMENT CAPABILITIES (EPB_PM_CAPABILITY, OFFSET 0xB2)**Note:** For BCM5708 only.**Table 457: PCI Power Management Capabilities (epb_pm_capability, Offset 0xb2)**

Bit	Field	Description	Mode	Reset	08
15	PME_IN_D3_COLD	When this bit is set it indicates that the device supports asserting PME# from the D3cold power state.	RO	1	X
14	PME_IN_D3_HOT	When this bit is set it indicates that the device supports asserting PME# from the D3hot power state.	RO	1	X
13	PME_IN_D2	When this bit is set it indicates that the device supports asserting PME# from the D2 state.	RO	0	X
12	PME_IN_D1	When this bit is set it indicates that the device supports asserting PME# from the D1 state.	RO	0	X
11	PME_IN_D0	When this bit is set it indicates that the device supports asserting PME# from the D0 state.	RO	1	X
10	D2_SUPPORT	When this bit is set it indicates that the device supports the D2 state.	RO	0	X
9	D1_SUPPORT	When this bit is set it indicates that the device supports the D1 state.	RO	0	X
8–6	AUX_CURRENT	This field is hardwired to the value 0x7 to indicate that the max AUX current requirement is 375 mA.	RO	0x7	X
5–2	RESERVED		RO	0x0	X
1–0	VERSION	These bits indicate that the device complies with revision 1.1 of the PCI Power Management specification.	RO	0x2	X

PCI POWER MANAGEMENT CONTROL/STATUS REGISTER (EPB_PM_CSR, OFFSET 0XB4)

Note: For BCM5708 only.

Table 458: PCI Power Management Control/Status Register (epb_pm_csr, Offset 0xb4)

Bit	Field	Description	Mode	Reset	06	08	09
15	PME_STATUS	This bit is hardwired to 0.	RO	0	X	X	-
14–13	DATA_SCALE (BCM5708 B0 and later)	These bits indicate the scaling factor to be used when interpreting the values returned by the “Power Management Data Register (epb_pm_data, Offset 0xb7)” on page 613.	RO	0	X	X	-
12–9	DATA_SEL (BCM5708 B0 and later)	These bits select which data is to be reported through the “Power Management Data Register (epb_pm_data, Offset 0xb7)” on page 613. Selecting values other than those listed will cause the epb_pm_data register to return 0.	RO	0	-	-	X
Value Name Description							
	0	0	This selects the power consumed in D0 state.				
	1	1	This selects the power consumed in D1 state.				
	2	2	This selects the power consumed in D2 state.				
	3	3	This selects the power consumed in D3 state.				
	4	4	This selects the power consumed in D0 state.				
	5	5	This selects the power consumed in D1 state.				
	6	6	This selects the power consumed in D2 state.				
	7	7	This selects the power consumed in D3 state.				
12–9	RESERVED		RO	0	X	X	-
8	PME_ENABLE	When this bit is set it enables PME# assertion (WAKE# or beacon).	RW	0	X	X	-
7–0	RESERVED		RO	0	X	X	-

POWER MANAGEMENT DATA REGISTER (EPB_PM_DATA, OFFSET 0xB7)**Note:** For BCM5708 only.

(BCM5708 B0 and later) The read-only Power Management Data Register returns one of eight different values selected by the DATA_SEL field of the “Power Management Data Register (epb_pm_data, Offset 0xb7)” on page 613 and has the value of 0 at power-on reset. The data returned by this register may be modified by firmware by writing to the “PCI Power Management Data 0 Register (pcipm_data_reg_0, offset 0x210)” on page 624 and “PCI Power Management Data Register 1 (pcipm_data_reg_1, offset 0x214)” on page 625.

PCI EXPRESS ENHANCED CAPABILITY HEADER REGISTER (EPB_ADVERR_CAPABILITY, OFFSET 0x100)**Note:** For BCM5708 only.**Table 459: PCI Express Enhanced Capability Header Register (epb_adVERR_capability, Offset 0x100)**

Bit	Field	Description	Mode	Reset	08
31–20	NEXT_CAP	This field points to the next PCI Express enhanced capability structure.	RO	0x14c	X
19–16	CAP_VER	This field returns the capability ID Version.	RO	0x1	X
15–0	CAP_ID	This field returns the capability ID of this capability structure and is hardwired to 0x0001 to indicate the presence of the PCI Express advanced error capability.	RW	0x0001	X

**PCI EXPRESS ADVANCED UNCORRECTABLE ERROR STATUS REGISTER
(EPB_ADVERR_UNCORR_ERROR_STATUS, OFFSET 0x104)**



Note: For BCM5708 only.

Table 460: PCI Express Advanced Uncorrectable Error Status Register (epb_adVERR_uncorr_error_status, Offset 0x104)

Bit	Field	Description	Mode	Reset	08
31–21	RESERVED		RO	0x000	X
20	UNSUPP_REQ_ERR_STA	This bit is set when the EPB detects an unsupported request.	WC	0	X
19	RESERVED		RO	0	X
18	MALFORMED_TLP_STA	This bit is set when the EPB detects a malformed TLP.	WC	0	X
17	RCVR_OVERFLOW_STA	This bit is set when the EPB detects a receiver overflow error.	WC	0	X
16	UNEXPECTED_COMP_STA	This bit is set when the EPB detects an unexpected completion.	WC	0	X
15	COMP_ABORT_STA	This bit is set when the EPB detects a completer abort.	WC	0	X
14	COMP_TIMEOUT_STA	This bit is set when the EPB detects a completer timeout.	WC	0	X
13	FC_PROTO_ERR_STA	This bit is set when the EPB detects a flow control protocol error.	WC	0	X
12	CORRUPT_TLP_STA	This bit is set when the EPB detects a corrupted TLP.	WC	0	X
11–5	RESERVED		RO	0	X
4	DLP_ERR_STA	This bit is set when the EPB detects a data link protocol error.	WC	0	X
3–1	RESERVED		RO	0	X
0	TRAINING_ERR_STA	This bit is set when the EPB detects a link training error.	WC	0	X

PCI EXPRESS ADVANCED UNCORRECTABLE ERROR MASK REGISTER (EPB_ADVERR_UNCORR_ERROR_MASK, OFFSET 0x108)



Note: For BCM5708 only.

Table 461: PCI Express Advanced Uncorrectable Error Mask Register (epb_adVERR_uncorr_error_mask, Offset 0x108)

Bit	Field	Description	Mode	Reset	08
31–21	RESERVED		RO	0x000	X
20	UNSUPP_REQ_ERR_MASK	When this bit is set the EPB logs unsupported request errors.	RWS	0	X
19	RESERVED		RO	0	X
18	MALFORMED_TLP_MASK	When this bit is set the EPB logs malformed TLP errors.	RWS	0	X
17	RCVR_OVERFLOW_MASK	When this bit is set the EPB logs receiver overflow errors.	RWS	0	X
16	UNEXPECTED_COMP_MASK	When this bit is set the EPB logs unexpected completion errors.	RWS	0	X
15	COMP_ABORT_MASK	When this bit is set the EPB logs completer abort errors.	RWS	0	X
14	COMP_TIMEOUT_MASK	When this bit is set the EPB logs completer timeout errors.	RWS	0	X
13	FC_PROTO_ERR_MASK	When this bit is set the EPB logs flow control protocol errors.	RWS	0	X
12	CORRUPT_TLP_MASK	When this bit is set the EPB logs corrupt TLP errors.	RWS	0	X
11–5	RESERVED		RO	0	X
4	DLP_ERR_MASK	When this bit is set the EPB logs data link protocol errors.	RWS	0	X
3–1	RESERVED		RO	0	X
0	TRAINING_ERR_MASK	When this bit is set the EPB logs link training errors.	RWS	0	X

**PCI EXPRESS ADVANCED UNCORRECTABLE ERROR SEVERITY REGISTER
(EPB_ADVERR_UNCORR_ERROR_SEVERITY, OFFSET 0x108)**



Note: For BCM5708 only.

Table 462: PCI Express Advanced Uncorrectable Error Severity Register (epb_adverr_uncorr_error_severity, Offset 0x108)

Bit	Field	Description	Mode	Reset	08
31–21	RESERVED		RO	0x000	X
20	UNSUPP_REQ_ERR_SEV	When this bit is set the EPB reports unsupported request errors as a fatal error to the root complex.	RWS	0	X
19	RESERVED		RO	0	X
18	MALFORMED_TLP_MASK	When this bit is set the EPB reports malformed TLP errors as a fatal error to the root complex.	RWS	1	X
17	RCVR_OVERFLOW_MASK	When this bit is set the EPB reports receiver overflow errors as a fatal error to the root complex.	RWS	1	X
16	UNEXPECTED_COMP_MASK	When this bit is set the EPB reports unexpected completion errors as a fatal error to the root complex.	RWS	0	X
15	COMP_ABORT_MASK	When this bit is set the EPB reports completer abort errors as a fatal error to the root complex.	RWS	0	X
14	COMP_TIMEOUT_MASK	When this bit is set the EPB reports completer timeout errors as a fatal error to the root complex.	RWS	0	X
13	FC_PROTO_ERR_MASK	When this bit is set the EPB reports flow control protocol errors as a fatal error to the root complex.	RWS	1	X
12	CORRUPT_TLP_MASK	When this bit is set the EPB reports corrupt TLP errors as a fatal error to the root complex.	RWS	0	X
11–5	RESERVED		RO	0	X
4	DLP_ERR_MASK	When this bit is set the EPB reports data link protocol errors as a fatal error to the root complex.	RWS	1	X
3–1	RESERVED		RO	0	X
0	TRAINING_ERR_MASK	When this bit is set the EPB reports link training errors as a fatal error to the root complex.	RWS	1	X

PCI EXPRESS ADVANCED CORRECTABLE ERROR STATUS REGISTER (EPB_ADVERR_CORR_ERROR_STA, OFFSET 0x110)



Note: For BCM5708 only.

Table 463: PCI Express Advanced Correctable Error Status Register (epb_adVERR_corr_error_sta, Offset 0x110)

Bit	Field	Description	Mode	Reset	08
31–13	RESERVED		RO	0x0000	X
12	REPLAY_TIMER_TIMEOUT_STA	This bit is set when the EPB detects a replay timer timeout.	WC	0	X
11–9	RESERVED		RO	0x0	X
8	REPLAY_NUM_ROLLOVER_STA	This bit is set when the EPB detects a replay number rollover.	WC	0	X
7	BAD_DLLP_STA	This bit is set when the EPB detects a bad DLLP.	WC	0	X
6	BAD_TLP_STA	This bit is set when the EPB detects a bad TLP.	WC	0	X
5–1	RESERVED		RO	0x00	X
0	RCVR_ERR_STA	This bit is set when the EPB detects a receiver error.	WC	0	X

PCI EXPRESS ADVANCED CORRECTABLE ERROR MASK REGISTER (EPB_ADVERR_CORR_ERROR_MASK, OFFSET 0x114)



Note: For BCM5708 only.

Table 464: PCI Express Advanced Correctable Error Mask Register (epb_adVERR_corr_error_mask, Offset 0x114)

Bit	Field	Description	Mode	Reset	08
31–13	RESERVED		RO	0x0000	X
12	REPLAY_TIMER_TIMEOUT_MASK	When this bit is set the EPB logs replay timer timeouts.	RWS	0	X
11–9	RESERVED		RO	0x0	X
8	REPLAY_NUM_ROLLOVER_MASK	When this bit is set the EPB logs replay number rollovers.	RWS	0	X
7	BAD_DLLP_MASK	When this bit is set the EPB logs bad DLLPs.	RWS	0	X
6	BAD_TLP_MASK	When this bit is set the EPB logs bad TLPs.	RWS	0	X
5–1	RESERVED		RO	0x00	X
0	RCVR_ERR_MASK	When this bit is set the EPB logs receiver errors.	RWS	0	X

PCI EXPRESS ADVANCED CAPABILITIES AND CONTROL REGISTER (EPB_ADV_CAP_CONTROL, OFFSET 0x118)



Note: For BCM5708 only.

Table 465: PCI Express Advanced Capabilities and Control Register (epb_adv_cap_control, Offset 0x118)

Bit	Field	Description	Mode	Reset	08
31–5	RESERVED		RO	0x000000	X
4–0	FIRST_ERR_PTR	This field points to the bit position in the “PCI Express Advanced Uncorrectable Error Status Register (epb_adverr_uncorr_error_status, Offset 0x104)” on page 614 that identifies the type of uncorrectable error captured in the “PCI Express Header Log Register (epb_header_log, Offset 0x11c)” on page 618.	RO	0	X

PCI EXPRESS HEADER LOG REGISTER (EPB_HEADER_LOG, OFFSET 0x11C)



Note: For BCM5708 only.

The read-only PCI Express Header Log register returns the header of the TLP associated with the uncorrectable error pointed to by the FIRST_ERR_PTR field of the “PCI Express Advanced Capabilities and Control Register (epb_adv_cap_control, Offset 0x118)” on page 618, is 4 DWORDs in length and has a value of 0 at power-on reset.

PIB UNCORRECTABLE ERROR STATUS REGISTER (EPB_PIB_UNCORR_ERROR_STATUS, OFFSET 0x12c)



Note: For BCM5708 only.

Table 466: PIB Uncorrectable Error Status Register (epb_pib_uncorr_error_status, Offset 0x12c)

Bit	Field	Description	Mode	Reset	08
31–14	RESERVED		RO	0x00000	
13	INT_BRIDGE_ERR_STA	This bit is set when an internal bridge error occurs on the PIB.	WC	0	X
12	SERR_ASSERT_STA	This bit is set when an SERR assertion occurs on the PIB.	WC	0	X
11	PERR_ASSERT_STA	This bit is set when a PERR assertion occurs on the PIB.	WC	0	X
10	DELAYED_TRANS_DISCARD_TIMER_STA	This bit is set when a delayed transaction discard timer expiration occurs on the PIB.	WC	0	X
9	UNCORR_ADDR_ERR_STA	This bit is set when an uncorrectable address error occurs on the PIB.	WC	0	X
8	UNCORR_ATTR_ERR_STA	This bit is set when an uncorrectable attribute error occurs on the PIB.	WC	0	X
7	UNCORR_DATA_ERR_STA	This bit is set when an uncorrectable data error occurs on the PIB.	WC	0	X
6	UNCORR_SPLIT_COMP_ERR_STA	This bit is set when an uncorrectable split completion message data error occurs on the PIB.	WC	0	X
5	UNEXPECTED_SPLIT_COMP_ERR_STA	This bit is set when an unexpected split completion error occurs on the PIB.	WC	0	X
4	RESERVED		RO	0	X
3	RCVD_MASTER_ABORT_STA	This bit is set when a received master abort error occurs on the PIB.	WC	0	X
2	RCVD_TARGET_ABORT_STA	This bit is set when a received target abort error occurs on the PIB.	WC	0	X
1	MASTER_ABORT_SPLIT_COMP_STA	This bit is set when a master abort on split completion error occurs on the PIB.	WC	0	X
0	TARGET_ABORT_SPLIT_COMP_STA	This bit is set when a target abort on split completion error occurs on the PIB.	WC	0	X

PIB UNCORRECTABLE ERROR MASK REGISTER (EPB_PIB_UNCORR_ERROR_MASK, OFFSET 0x130)



Note: For BCM5708 only.

Table 467: PIB Uncorrectable Error Mask Register (epb_pib_uncorr_error_mask, Offset 0x130)

Bit	Field	Description	Mode	Reset	08
31–14	RESERVED		RO	0x00000	X
13	INT_BRIDGE_ERR_MASK	When this bit is set the PIB logs internal bridge errors.	RWS	0	X
12	SERR_ASSERT_MASK	When this bit is set the PIB logs SERR assertion.	RWS	1	X
11	PERR_ASSERT_MASK	When this bit is set the PIB logs PERR assertion.	RWS	0	X
10	DELAYED_TRANS_DISCARD_TIMER_MASK	When this bit is set the PIB logs delayed transaction discard timer expiration errors.	RWS	1	X
9	UNCORR_ADDR_ERR_MASK	When this bit is set the PIB logs uncorrectable address errors.	RWS	1	X
8	UNCORR_ATTR_ERR_MASK	When this bit is set the PIB logs uncorrectable attribute errors.	RWS	1	X
7	UNCORR_DATA_ERR_MASK	When this bit is set the PIB logs uncorrectable data errors.	RWS	1	X
6	UNCORR_SPLIT_COMP_ERR_MASK	When this bit is set the PIB logs uncorrectable split completion messages.	RWS	1	X
5	UNEXPECTED_SPLIT_COMP_ERR_MASK	When this bit is set the PIB logs unexpected split completion errors.	RWS	0	X
4	RESERVED		RO	0	X
3	RCVD_MASTER_ABORT_MASK	When this bit is set the PIB logs received master abort errors.	RWS	1	X
2	RCVD_TARGET_ABORT_MASK	When this bit is set the PIB logs received target abort errors.	RWS	0	X
1	MASTER_ABORT_SPLIT_COMP_MASK	When this bit is set the PIB logs master abort on split completion errors.	RWS	0	X
0	TARGET_ABORT_SPLIT_COMP_MASK	When this bit is set the PIB logs target abort on split completion errors.	RWS	0	X

PIB UNCORRECTABLE ERROR SEVERITY REGISTER (EPB_PIB_UNCORR_ERROR_SEV, OFFSET 0x134)



Note: For BCM5708 only.

Table 468: PIB Uncorrectable Error Severity Register (epb_pib_uncorr_error_sev, Offset 0x134)

Bit	Field	Description	Mode	Reset	08
31–14	RESERVED		RO	0x00000	X
13	INT_BRIDGE_ERR_SEV	When this bit is set the PIB reports an internal bridge error as a fatal error to the root complex.	RWS	0	X
12	SERR_ASSERT_SEV	When this bit is set the PIB reports a SERR assertion as a fatal error to the root complex.	RWS	1	X
11	PERR_ASSERT_SEV	When this bit is set the PIB reports a PERR assertion as a fatal error to the root complex.	RWS	0	X
10	DELAYED_TRANS_DISCARD_TIMER_SEV	When this bit is set the PIB reports a delayed transaction discard timer expiration as a fatal error to the root complex.	RWS	0	X
9	UNCORR_ADDR_ERR_SEV	When this bit is set the PIB reports an uncorrectable address error as a fatal error to the root complex.	RWS	1	X
8	UNCORR_ATTR_ERR_SEV	When this bit is set the PIB reports an uncorrectable attribute error as a fatal error to the root complex.	RWS	1	X
7	UNCORR_DATA_ERR_SEV	When this bit is set the PIB reports an uncorrectable data error as a fatal error to the root complex.	RWS	0	X
6	UNCORR_SPLIT_COMP_ERR_SEV	When this bit is set the PIB reports an uncorrectable split completion message data error as a fatal error to the root complex.	RWS	1	X
5	UNEXPECTED_SPLIT_COMP_ERR_SEV	When this bit is set the PIB reports an unexpected split completion error as a fatal error to the root complex.	RWS	0	X
4	RESERVED		RO	0	X
3	RCVD_MASTER_ABORT_SEV	When this bit is set the PIB reports a received master abort error as a fatal error to the root complex.	RWS	0	X
2	RCVD_TARGET_ABORT_SEV	When this bit is set the PIB reports a received target abort error as a fatal error to the root complex.	RWS	0	X
1	MASTER_ABORT_SPLIT_COMP_SEV	When this bit is set the PIB reports a master abort on split completion error as a fatal error to the root complex.	RWS	0	X
0	TARGET_ABORT_SPLIT_COMP_SEV	When this bit is set the PIB reports a target abort on split completion error as a fatal error to the root complex.	RWS	0	X

PIB ERROR CAPABILITIES AND CONTROL REGISTER (EPB_PIB_ERROR_CAP_CONTROL, OFFSET 0x138)



Note: For BCM5708 only.

Table 469: PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)

Bit	Field	Description	Mode	Reset	08
31–5	RESERVED		RO	0x0000000	X
4–0	FIRST_ERR_PTR	This field points to the bit position in the “PIB Uncorrectable Error Status Register (epb_pib_uncorr_error_status, Offset 0x12c)” on page 619 that identifies the type of uncorrectable error captured in the “PIB Header Log Register (epb_pib_header_log, Offset 0x13c)” on page 622.	RO	0x1	X

PIB HEADER LOG REGISTER (EPB_PIB_HEADER_LOG, OFFSET 0x13c)



Note: For BCM5708 only.

Table 470: PIB Header Log Register (epb_pib_header_log, Offset 0x13c)

Bit	Field	Description	Mode	Reset	08
127–96	SECOND_ADDR_PHASE	This field logs the second address phase of the error logged by the FIRST_ERR_PTR field of the “PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)” on page 622.	RO	0x000000000	X
95–64	FIRST_ADDR_PHASE	This field logs the first address phase of the error logged by the FIRST_ERR_PTR field of the “PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)” on page 622.	RO	0x000000000	X
63–44	RESERVED		RO	0x0000000	X
43–40	SECOND_CBE_DAC	This field logs the CBE[3:0] bits during the second phase of a dual-address cycle of the error logged by the FIRST_ERR_PTR field of the “PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)” on page 622.	RO	0x0	X
39–36	FIRST_CBE_DAC	This field logs the CBE[3:0] bits during the first phase of a dual-address cycle of the error logged by the FIRST_ERR_PTR field of the “PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)” on page 622.	RO	0x0	X
35–32	CBE	This field logs the CBE[3:0] bits during the attribute phase of the error logged by the FIRST_ERR_PTR field of the “PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138)” on page 622.	RO	0x0	X

Table 470: PIB Header Log Register (epb_pib_header_log, Offset 0x13c) (Cont.)

Bit	Field	Description	Mode	Reset	08
31–0	ADDRESS	This field logs the AD[31:0] bits during the attribute phase of the error logged by the FIRST_ERR_PTR field of the “ PIB Error Capabilities and Control Register (epb_pib_error_cap_control, Offset 0x138) ” on page 622.	RO	0x00000000	X

POWER BUDGETING ENHANCED CAPABILITY HEADER REGISTER (EPB_POWER_CAPABILITY, OFFSET 0x14c)



Note: For BCM5708 only.

Table 471: Power Budgeting Enhanced Capability Header Register (epb_power_capability, Offset 0x14c)

Bit	Field	Description	Mode	Reset	08
31–20	NEXT_CAP	This field points to the next PCI Express enhanced capability structure.	RO	0x0	X
19–16	CAP_VER	This field returns the capability ID version.	RO	0x1	X
15–0	CAP_ID	This field returns the capability ID of this capability structure and is hardwired to 0x0004 to indicate the presence of the PCI Express power budgeting capability.	RW	0x0004	X

POWER BUDGETING DATA SELECT REGISTER (EPB_PB_DS, OFFSET 0x150)



Note: For BCM5708 only.

Table 472: Power Budgeting Data Select Register (epb_pb_ds, Offset 0x150)

Bit	Field	Description	Mode	Reset	08
31–8	RESERVED		RO	0x0000000	X
7–0	DATA_SELECT	When this field is written it acts as an index for access to the “Power Budgeting Data Register (epb_pb_dr, Offset 0x154)” on page 624.	RW	0x00	X

POWER BUDGETING DATA REGISTER (EPB_PB_DR, OFFSET 0x154)



Note: For BCM5708 only.

Table 473: Power Budgeting Data Register (epb_pb_dr, Offset 0x154)

Bit	Field	Description	Mode	Reset	08
31–0	RESERVED		RO	0x000000000	X

POWER BUDGETING CAPABILITY REGISTER (EPB_PB_CAP, OFFSET 0x158)

Table 474: Power Budgeting Capability Register (epb_pb_cap, Offset 0x158)

Bit	Field	Description	Mode	Reset	08
31–0	RESERVED		RO	0x000000000	X

PCI POWER MANAGEMENT DATA 0 REGISTER (PCIPM_DATA_REG_0, OFFSET 0x210)



Note: For BCM5708 only.

(BCM5708 B0 and later) The read/write PCI Power Management Data 0 register is only accessible by firmware and is used to set the power consumed values returned by the “Power Management Data Register (epb_pm_data, Offset 0xb7)” on page 613.

Table 475: PCI Power Management Data 0 Register (pcipm_data_reg_0)

Bit	Name	Description	Mode	Reset	08
23–24	PC_D3	Power consumed in D3 state.	RW	0	X
23–16	PC_D2	Power consumed in D2 state.	RW	0	X
15–8	PC_D1	Power consumed in D1 state.	RW	0	X
7–0	PC_D0	Power consumed in D0 state.	RW	0	X

PCI POWER MANAGEMENT DATA REGISTER 1 (PCIPM_DATA_REG_1, OFFSET 0x214)

Note: For BCM5708 only.

(BCM5708 B0 and later) The read/write PCI Power Management Data 1 register is only accessible by firmware and is used to set the power dissipated values returned by the “[Power Management Data Register \(epb_pm_data, Offset 0xb7\)](#)” on page [613](#)“.

Table 476: PCI Power Management Data 1 Register (pcipm_data_reg_1)

Bit	Name	Description	Mode	Reset	08
31–24	PC_D3	Power dissipated in D3 state.	RW	0	X
23–16	PC_D2	Power dissipated in D2 state.	RW	0	X
15–8	PC_D1	Power dissipated in D1 state.	RW	0	X
7–0	PC_D0	Power dissipated in D0 state.	RW	0	X

Section 12: BCM5706C, BCM5706S, and BCM5708C Transceiver Register Summary

MII MANAGEMENT INTERFACE REGISTER PROGRAMMING

Access to the following transceiver registers is provided indirectly through the MII Communication Register (see “[EMAC MDIO Command Register \(emac_mdio_comm, Offset 0x14ac\)](#)” on page 321) of the BCM5706C, BCM5706S, and BCM5708C controllers using a PHY_ADDR field value of 1h. The BCM5706C and BCM5708C support the 1000BASE-T registers (also known as Copper registers) while the BCM5706S supports the 1000BASE-X registers (also known as Fiber or SerDes registers).

REGISTER MAP

The NetXtreme II transceiver contains the following set of registers.

Table 477: Register Map

Address	Register Table
1000BASE-T/100BASE-TX/10BASE-T Registers	
00h	Table 550: “1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h),” on page 737
01h	Table 551: “1000BASE-T/100BASE-TX/10BASE-TMII Status Register (Address 01h),” on page 739
02h-03h	Table 552: “1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register MSB (Address 02h),” on page 742
04h	Table 554: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h),” on page 743
05h	Table 555: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register (Address 05h),” on page 746
06h	Table 556: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h),” on page 749
07h	Table 557: “1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h),” on page 751
08h	Table 558: “1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h),” on page 752
09h	Table 559: “1000BASE-T Control Register (Address 09h),” on page 753
0Ah	Table 560: “1000BASE-T Status Register (Address 0Ah),” on page 755
0Ch-0Eh	Reserved (Do not read from or write to a reserved register.)
0Fh	Table 561: “1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh),” on page 757
10h	Table 562: “1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h),” on page 758
11h	Table 563: “1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h),” on page 760

Table 477: Register Map (Cont.)

Address	Register Table
12h	Table 564: “1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h),” on page 763
13h	Table 565: “1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h),” on page 764
14h	Table 567: “1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h),” on page 766
15h-16h	Reserved (Do not read from or write to a reserved register except for accessing the Expansion registers through register 15h.)
17h	Table 569: “Expansion Register Access Register (Address 17h),” on page 768
18h	Table 574: “1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow 0h),” on page 771 Table 575: “10BASE-T Register (Address 18h, Shadow 1h),” on page 773 Table 576: “1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow 2h),” on page 775 Table 577: “1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow 4h),” on page 777 Table 578: “1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow 7h),” on page 779
19h	Table 579: “1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h),” on page 781
1Ah	Table 580: “1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah),” on page 784
1Bh	Table 581: “1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh),” on page 787

Table 477: Register Map (Cont.)

Address	Register Table
1Ch	<p>Table 515: “1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow Value 00010),” on page 677</p> <p>Table 516: “1000BASE-T/100BASE-TX/10BASE-T Clock Alignment Control Register (Address 1Ch, Shadow Value 00011),” on page 679</p> <p>Table 585: “1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow 04h),” on page 792</p> <p>Table 586: “1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h),” on page 794</p> <p>Table 587: “1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow 08h),” on page 796</p> <p>Table 588: “1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow 09h),” on page 798</p> <p>Table 589: “Auto Power-Down Register (Address 1Ch, Shadow 0Ah),” on page 802</p> <p>Table 592: “1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow 0Dh),” on page 806</p> <p>Table 593: “1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow 0Eh),” on page 808</p> <p>Table 594: “LED GPIO Control/Status Register (Address 1Ch, Shadow Value 0Fh),” on page 810</p> <p>Table 598: “1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h),” on page 816</p> <p>Table 526: “1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010),” on page 695</p> <p>Table 599: “Auxiliary Control Register (Address 1Ch, Shadow 1Bh),” on page 818</p> <p>Table 600: “Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow 1Ch),” on page 820</p> <p>Table 601: “Misc. 1000BASE-X Status Register (Address 1Ch, Shadow 1Dh),” on page 822</p> <p>Table 530: “1000BASE-T/100BASE-TX/10BASE-T Autodetect Medium Register (Address 1Ch, Shadow Value 11110),” on page 704</p> <p>Table 602: “Mode Control Register (Address 1Ch, Shadow 1Fh),” on page 824</p>
1Dh	<p>Table 603: “1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0,” on page 826</p> <p>Table 604: “1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1,” on page 828</p>
1Eh	Table 605: “Test Register 1 (Address 1Eh),” on page 831
1Fh	Reserved (Do not read from or write to a reserved register.)
1000BASE-X Registers (enabled by register 1Ch, shadow value 11111, bit 0 = 1)	
00h	Table 535: “1000BASE-X MII Control Register (Address 00h),” on page 713
01h	Table 536: “1000BASE-X MII Status Register (Address 01h),” on page 715
04h	Table 537: “1000BASE-X Auto-Negotiation Advertisement Register (Address 04h),” on page 717
05h	Table 539: “1000BASE-X Auto-Negotiation Link Partner Ability Register—Base Page (Address 05h) When in GMII/RGMII/RTBI Mode,” on page 720
06h	Table 541: “1000BASE-X Auto-Negotiation Extended Status Register (Address 06h),” on page 723
07h–0Eh	Reserved (Do not read from or write to reserved register)
0Fh	Table 542: “1000BASE-X IEEE Extended Status Register (Address 0Fh),” on page 724

Table 477: Register Map (Cont.)

Address	Register Table
Expansion Registers: Read/Write through Register 15h (accessed by writing to register 17h, bits [11:0] = 1111 + Expansion register number)	
00h	Table 543: "Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)," on page 725
01h	Table 544: "Expansion Register 01h: Expansion Interrupt Status Register (Address 15h)," on page 725
02h	Table 545: "Expansion Register 02h: Expansion Interrupt Mask Register (Address 15h)," on page 727
04h	Table 607: "Expansion Register 04h: Multicolor LED Selector Register (Address 15h)," on page 834
05h	Table 608: "Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)," on page 836
06h	Table 609: "Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)," on page 837

REGISTER NOTATIONS

In the register description tables, the following notation in the R/W column is used to describe the ability to read or write:

- R/W = Read or write
- RO = Read only
- LH = Latched high (until read)
- LL = Latched low (until read)
- H = Fixed high
- L = Fixed low
- SC = Self-clear after read
- CR = Clear on read

Reserved bits must be written as the default value and ignored when read.

1000BASE-T/100BASE-TX/10BASE-T REGISTERS DESCRIPTIONS

1000BASE-T/100BASE-TX/10BASE-T MII CONTROL REGISTER (ADDRESS 00H)

Table 478: 1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h)

Bit	Name	R/W	Description	Default
15	Reset	R/W SC	1 = PHY reset 0 = Normal operation	0
14	Internal Loopback	R/W	1 = Loopback mode 0 = Normal operation	0
13	Speed Selection (LSB)	R/W	Bits [6,13]: 1 1 = Reserved 1 0 = 1000 Mbps 0 1 = 100 Mbps 0 0 = 10 Mbps	0
12	Auto-Negotiation Enable	R/W	1 = Auto-negotiation enabled 0 = Auto-negotiation disabled	1
11	Power Down	R/W	1 = Power down 0 = Normal operation	0
10	Isolate	R/W	1 = Electrically isolate PHY from GMII/ RGMII/RTBI 0 = Normal operation	0
9	Restart Auto-Negotiation	R/W SC	1 = Restarting auto-negotiation 0 = Auto-negotiation restart complete	0
8	Duplex Mode	R/W	1 = Full-duplex 0 = Half-duplex	1
7	Collision Test Enable	R/W	1 = Enable the collision test mode 0 = Disable the collision test mode	0
6	Speed Selection (MSB)	R/W	Works in conjunction with bit 13	1
5	Reserved	R/W	Write as 0, ignore on read	0
4	Reserved	R/W	Write as 0 ignore on read	0
3	Reserved	R/W	Write as 0 ignore on read	0
2	Reserved	R/W	Write as 0 ignore on read	0
1	Reserved	R/W	Write as 0 ignore on read	0
0	Reserved	R/W	Write as 0 ignore on read	0

Reset

To reset the NetXtreme II by software control, a 1 must be written to bit 15 of the MII Control register. This bit clears itself after the reset process is complete, and does not need to be cleared using a second MII write. Writes to other MII Control register bits have no effect until the reset process is completed, which requires approximately 2.0 µs. Writing a 0 to this bit has no effect. A 1 is returned when this bit is read during the reset process; otherwise, it returns a 0.

Internal Loopback

The NetXtreme II can be placed into internal loopback mode by setting bit 14 of the MII Control register. The loopback mode can be cleared by writing a 0 to bit 14 of the MII Control register, or by resetting the chip. A 1 is returned when this bit is read and the chip is in loopback mode; otherwise, it returns a 0.

Speed Selection (LSB)

When auto-negotiation is disabled, bits 6 and 13 of the MII Control register can be used to manually select the speed of operation. When bit 6 is set and bit 13 is cleared, 1000BASE-T operation is selected. When bit 6 is cleared and bit 13 is set, 100BASE-TX operation is selected. When both bits are cleared, 10BASE-T operation is selected. Setting both bits is not permitted. When read, these bits return the last value written. The default value for bit 13 is determined by the state of the SPD0 and F1000 pins at reset.

Auto-Negotiation Enable

When bit 12 of the MII Control register is set, the NetXtreme II mode of operation is controlled by auto-negotiation. When this bit is cleared, the NetXtreme II mode of operation is determined by the Manual Speed, Duplex Mode, and Master/Slave Configuration bits. A 1 is returned when this bit is read with auto-negotiation enabled; otherwise, it returns a 0. The default value of this bit is determined by the state of the ANEN pin at reset.

Power-Down

When bit 11 of the MII Control register is set, the NetXtreme II is placed into low power standby mode. The default value of this bit will be set to a 1 if the PHY is configured for Fiber mode, INTF_SEL[3:2] = 01.

Isolate

The NetXtreme II can be isolated from the GMII/RGMII/RTBI bus by setting bit 10 of the MII Control register. All GMII/RGMII/RTBI outputs are tristated, and all GMII/RGMII/RTBI inputs are ignored. Because the management interface is still active, the isolate mode can be cleared by writing a 0 to bit 10 of the MII Control register or by resetting the chip. A 1 is returned when this bit is read and the chip is in isolate mode; otherwise, it returns a 0. This default of this bit is a 0.

Restart Auto-Negotiation

Setting bit 9 of the MII Control register forces the auto-negotiation process to be restarted, regardless of the current state of the auto-negotiation state machine. When auto-negotiation is enabled, setting this bit restarts the auto-negotiation process. Clearing this bit has no effect. This bit is self-clearing. After the auto-negotiation process has restarted, it returns a value of 0.

Duplex Mode

When auto-negotiation is disabled, the duplex mode of the NetXtreme II can be controlled by writing to bit 8 of the MII Control register. Setting this bit forces the NetXtreme II into full-duplex operation, and clearing this bit forces the NetXtreme II into half-duplex operation. When this bit is read, it returns the last value written. The default value of this bit is determined by the FDX pin at reset.

Speed Selection (MSB)

When auto-negotiation is disabled, bits 6 and 13 of the MII Control register can be used to manually select the speed of operation. When bit 6 is set and bit 13 is cleared, 1000BASE-T operation is selected. When bit 6 is cleared and bit 13 is set, 100BASE-TX operation is selected. When both bits are cleared, 10BASE-T operation is selected. Setting both bits is not permitted. When read, these bits return the last value written. The default value for bit 6 is determined by the state of the F1000 pins at reset.

Collision Test

The NetXtreme II can be placed into collision test mode by setting to bit 7 of the MII Control register. In this mode, the COL pin is asserted whenever the TX_EN pin is driven high. The collision test mode can be cleared by writing a 0 to bit 7 of the MII Control register, or by resetting the chip. A 1 is returned when this bit is read and the chip is in collision test mode; otherwise, it returns a 0.

1000BASE-T/100BASE-TX/10BASE-T MII STATUS REGISTER (ADDRESS 01H)

Table 479: 1000BASE-T/100BASE-TX/10BASE-TMII Status Register (Address 01h)

Bit	Name	R/W	Description	Default
15	100BASE-T4 Capable	RO L	1 = 100BASE-T4 capable 0 = Not 100BASE-T4 capable	0
14	100BASE-X Full-Duplex Capable	RO H	1 = 100BASE-X full-duplex capable 0 = Not 100BASE-X full-duplex capable	1
13	100BASE-X Half-Duplex Capable	RO H	1 = 100BASE-X half-duplex capable 0 = Not 100BASE-X half-duplex capable	1
12	10BASE-T Full-Duplex Capable	RO H	1 = 10BASE-T full-duplex capable 0 = Not 10BASE-T full-duplex capable	1
11	10BASE-T Half-Duplex Capable	RO H	1 = 10BASE-T half-duplex capable 0 = Not 10BASE-T half-duplex capable	1
10	100BASE-T2 Full-Duplex Capable	RO L	1 = 100BASE-T2 full-duplex capable 0 = Not 100BASE-T2 full-duplex capable	0
9	100BASE-T2 Half-Duplex Capable	RO L	1 = 100BASE-T2 half-duplex capable 0 = Not 100BASE-T2 half-duplex capable	0
8	Extended Status	RO H	1 = Extended status information in reg 0Fh 0 = No extended status information in reg 0Fh	1
7	Reserved	RO	Ignore on read	0
6	Management Frames Preamble Suppression	RO H	1 = Preamble can be suppressed 0 = Preamble always required	1
5	Auto-Negotiation Complete	RO	1 = Auto-negotiation complete 0 = Auto-negotiation in progress	0
4	Remote Fault	RO LH	1 = Remote fault detected 0 = No remote fault detected	0
3	Auto-Negotiation Ability	RO H	1 = Auto-negotiation capable 0 = Not auto-negotiation capable	1
2	Link Status	RO LL	1 = Link is up (Link Pass state) 0 = Link is down (Link Fail state)	0
1	Jabber Detect	RO LH	1 = Jabber condition detected 0 = No jabber condition detected	0
0	Extended Capability	RO H	1 = Extended register capabilities 0 = No extended register capabilities	1

100BASE-T4 Capable

The NetXtreme II is not capable of 100BASE-T4 operation, and returns a 0 when bit 15 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

100BASE-X Full-Duplex Capable

The NetXtreme II is capable of 100BASE-TX full-duplex operation, and returns a 1 when bit 14 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

100BASE-X Half-Duplex Capable

The NetXtreme II is capable of 100BASE-X half-duplex operation, and returns a 1 when bit 13 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

10BASE-T Full-Duplex Capable

The NetXtreme II is capable of 10BASE-T full-duplex operation, and returns a 1 when bit 12 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

10BASE-T Half-Duplex Capable

The NetXtreme II is capable of 10BASE-T half-duplex operation, and returns a 1 when bit 11 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

100BASE-T2 Full-Duplex Capable

The NetXtreme II is not capable of 100BASE-T2 full-duplex operation, and returns a 0 when bit 10 is read.

100BASE-T2 Half-Duplex Capable

The NetXtreme II is not capable of 100BASE-T2 half-duplex operation, and returns a 0 when bit 9 is read.

Extended Status

The NetXtreme II contains IEEE Extended Status register at address 0Fh, and returns a 1 when bit 8 is read.

Management Frames Preamble Suppression

The NetXtreme II accepts MII management frames whether or not they are preceded by the preamble pattern, and returns a 1 when bit 6 is read.



Note: Preamble is still required on the first read or write. There is no method to disable Preamble Suppression.

Auto-Negotiation Complete

The NetXtreme II returns a 1 in bit 5 and the contents of registers 4, 5, and 6 are valid. This bit returns a 0 while auto-negotiation is in progress.

Remote Fault

The NetXtreme II returns a 1 in bit 4 when its link partner has signaled a remote fault condition. When a remote fault occurs, the bit is set and remains so until the remote fault condition has been cleared and the register is read.

Auto-Negotiation Ability

Even if the auto-negotiation function has been disabled, the NetXtreme II is capable of performing IEEE Auto-Negotiation and returns a 1 when bit 3 is read.

Link Status

The NetXtreme II returns a 1 in bit 2 when the link monitor is in the link pass state (indicating that a valid link has been established), otherwise it returns a 0. When a link failure occurs, the Link Status bit is latched at 0 and remains so until the bit is read and the NetXtreme II is in the link pass state.

Jabber Detect

Jabber detection is performed within the PHY and the result is latched into this bit. When a jabber condition has been detected, the NetXtreme II returns a 1 in bit 1. The bit is cleared by reading.

Extended Capability

The NetXtreme II supports Extended Capability registers, and returns a 1 when bit 0 is read.

1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register (Address 02h and 03h)

Table 480: 1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register (Addresses 02h and 03h)

Bit	Name	R/W	Description	Default	06	08
15:0	Address 02: ID MSBs	RO	16 MSBs of PHY Identifier	0020 (hex)	X	X
15:0	Address 03: ID LSBs	RO	16 LSBs of PHY Identifier	615n ^a (hex)	X	-
15:0	Address 03: ID LSBs	RO	16 LSBs of PHY Identifier	636n ^a (hex)	-	X

a = The revision number (*n*) changes with each silicon revision.

The IEEE has issued an Organizationally Unique Identifier (OUI) to Broadcom Corporation. This 24-bit number allows devices made by Broadcom to be distinguished from all other manufacturers. The OUI combined with model numbers and revision numbers assigned by Broadcom precisely identifies a device manufactured by Broadcom.

The [15:0] bits of MII register 02h (PHYID HIGH) contain OUI bits [3:18]. The [15:0] bits of MII register 03h (PHYID LOW) contain the most significant OUI bits [19:24], 6 Manufacturer's Model Number bits, and 4 Revision Number bits. The 2 least significant OUI binary bits are not used.

Broadcom Corporation's OUI is 00-10-18, expressed as hexadecimal values. The binary OUI is 0000-0000-0000-1000-0001-1000. The model number for the NetXtreme II is 0Bh. Revision numbers start with 0h and increment by 1 for each chip modification.

- PHYID HIGH[15:0] = OUI[3:18]
- PHYID LOW[15:0] = OUI[19:24] + Model[5:0] + Revision [3:0]



1000BASE-T/100BASE-TX/10BASE-T AUTO-NEGOTIATION ADVERTISEMENT REGISTER (ADDRESS 04H)

Table 481: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h)

Bit	Name	R/W	Description	Default
15	Next Page	R/W	1 = Next page ability supported 0 = Next page ability not supported	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Remote Fault	R/W	1 = Advertise remote fault detected 0 = Advertise no remote fault detected	0
12	Reserved Technology	R/W	Write as 0, ignore on read	0
11	Asymmetric Pause	R/W	1 = Advertise asymmetric pause 0 = Advertise no asymmetric pause	0
10	Pause Capable	R/W	1 = Capable of full-duplex pause operation 0 = Not capable of pause operation	0
9	100BASE-T4 Capable	R/W	1 = 100BASE-T4 capable 0 = Not 100BASE-T4 capable	0
8	100BASE-TX Full-Duplex Capable	R/W	1 = 100BASE-TX full-duplex capable 0 = Not 100BASE-TX full-duplex capable	1
7	100BASE-TX Half-Duplex Capable	R/W	1 = 100BASE-TX half-duplex capable 0 = Not 100BASE-TX half-duplex capable	1
6	10BASE-T Full-Duplex Capable	R/W	1 = 10BASE-T full-duplex capable 0 = Not 10BASE-T full-duplex capable	1
5	10BASE-T Half-Duplex Capable	R/W	1 = 10BASE-T half-duplex capable 0 = Not 10BASE-T half-duplex capable	1
4	Selector Field	R/W	Bits[4:0] = 00001 indicates IEEE 802.3 CSMA/CD	0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		1

Next Page

Bit 15 must be set = 1 when the management software wants to control Next Page exchange. When this bit is cleared, Next Page exchange is controlled automatically by the NetXtreme II. When this bit is cleared and the NetXtreme II is not advertising 1000BASE-T capability, no Next Page exchange occurs.

Remote Fault

Setting bit 13 sends a remote fault indication to the link partner during auto-negotiation. Writing a 0 to this bit clears the Remote Fault transmission bit. This bit returns a 1 when advertising remote fault; otherwise, it returns a 0.

Reserved Technology

Bit 12 is reserved for future versions of the auto-negotiation standard, and must always be written as 0.

Asymmetric Pause

When Bit 11 is set = 1, the NetXtreme II advertises that asymmetric pause is preferred. When the bit is cleared, the NetXtreme II advertises that asymmetric pause is not needed. This bit returns a 1 when advertising asymmetric pause; otherwise, it returns a 0. When advertising asymmetric pause, bit 10 indicates the preferred direction of the pause operation. Setting bit 10 indicates that pause frames flow toward the NetXtreme II. Clearing bit 10 indicates that pause frames flow toward the link partner.

Pause Capable

When Bit 10 is set = 1, the NetXtreme II advertises full-duplex pause capability. When the bit is cleared, the NetXtreme II advertises no pause capability. This bit returns a 1 when advertising pause capability; otherwise, it returns a 0.

100BASE-T4 Capable

The NetXtreme II does not support 100BASE-T4 capability. Do not write a 1 to bit 9.

100BASE-TX Full-Duplex Capable

When bit 8 is set = 1, the NetXtreme II advertises 100BASE-TX full-duplex capability. When the bit is cleared, the NetXtreme II advertises no 100BASE-TX full-duplex capability. This bit returns a 1 when advertising 100BASE-TX full-duplex capability; otherwise, it returns a 0.

F1000	SPD0	FDX	Bit 7 Default
1	0	1	1
0	1	1	1
1	0	0	1
0	1	0	1
Note: All other combinations default to 0.			0

10BASE-T Full-Duplex Capable

When bit 6 is set = 1, the NetXtreme II advertises 10BASE-T full-duplex capability. When the bit is cleared, the NetXtreme II advertises no 10BASE-T full-duplex capability. This bit returns a 1 when advertising 10BASE-T full-duplex capability; otherwise, it returns a 0.

10BASE-T Half-Duplex Capable

When bit 5 is set = 1, the NetXtreme II advertises 10BASE-T half-duplex capability. When the bit is cleared, the NetXtreme II advertises no 10BASE-T half-duplex capability. This bit returns a 1 when advertising 10BASE-T half-duplex capability; otherwise, it returns a 0.

Selector Field

Bits [4:0] indicate the protocol type. The value 00001 indicates that the NetXtreme II belongs to the 802.3 class of PHY transceivers.

1000BASE-T/100BASE-TX/10BASE-T AUTO-NEGOTIATION LINK PARTNER REGISTER (ADDRESS 05h)

**Table 482: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register
(Address 05h)**

Bit	Name	R/W	Description	Default
15	Next Page	RO	1 = Link partner has Next Page ability 0 = Link partner does not have Next Page ability	0
14	Acknowledge	RO	1 = Link partner has received link code word 0 = Link partner has not received link code word	0
13	Remote Fault	RO	1 = Link partner has detected remote fault 0 = Link partner has not detected remote fault	0
12	Reserved Technology	RO	Write as 0, ignore on read	0
11	Asymmetric Pause	RO	1 = Link partner wants asymmetric pause 0 = Link partner does not want asymmetric pause	0
10	Pause Capable	RO	1 = Link partner is capable of pause operation 0 = Link partner is not capable of pause operation	0
9	100BASE-T4 Capable	RO	1 = Link partner is 100BASE-T4 capable 0 = Link partner is not 100BASE-T4 capable	0
8	100BASE-TX Full-Duplex Capable	RO	1 = Link partner is 100BASE-TX full-duplex capable 0 = Link partner is not 100BASE-TX full-duplex capable	0
7	100BASE-TX Half-Duplex Capable	RO	1 = Link partner is 100BASE-TX half-duplex capable 0 = Link partner not 100BASE-TX half-duplex capable	0
6	10BASE-T Full-Duplex Capable	RO	1 = Link partner is 10BASE-T full-duplex capable 0 = Link partner is not 10BASE-T full-duplex capable	0
5	10BASE-T Half-Duplex Capable	RO	1 = Link partner is 10BASE-T half-duplex capable 0 = Link partner is not 10BASE-T half-duplex capable	0
4	Protocol Selector Field	RO	Link partner protocol selector field	0
3		RO		0
2		RO		0
1		RO		0
0		RO		0



Note: As indicated by bit 5 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register, the values contained in the 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability register are only guaranteed to be valid after auto-negotiation has successfully completed.

Next Page

The NetXtreme II returns a 1 in bit 15 when the link partner wants to transmit Next Page information.

Acknowledge

The NetXtreme II returns a 1 in bit 14 when the link partner has acknowledged reception of the link code word; otherwise, it returns a 0.

Remote Fault

The NetXtreme II returns a 1 in bit 13 when the link partner has advertised detection of a remote fault; otherwise, it returns a 0.

Reserved Technology

Bit 12 is reserved for future versions of the auto-negotiation standard, and must be ignored when read.

Asymmetric Pause

The NetXtreme II returns a 1 in bit 11 when the link partner has advertised asymmetric pause; otherwise, it returns a 0.

Pause Capable

The NetXtreme II returns a 1 in bit 10 when the link partner has advertised Pause Capability; otherwise, it returns a 0.

100BASE-T4 Capable

The NetXtreme II returns a 1 in bit 9 when the link partner has advertised 100BASE-T4 capability; otherwise, it returns a 0.

100BASE-TX Full-Duplex Capable

The NetXtreme II returns a 1 in bit 8 when the link partner has advertised 100BASE-TX full-duplex capability; otherwise, it returns a 0.

100BASE-TX Half-Duplex Capable

The NetXtreme II returns a 1 in bit 7 when the link partner has advertised 100BASE-TX half-duplex capability; otherwise, it returns a 0.

10BASE-T Full-Duplex Capable

The NetXtreme II returns a 1 in bit 6 when the link partner has advertised 10BASE-T full-duplex capability; otherwise, it returns a 0.

10BASE-T Half-Duplex Capable

The NetXtreme II returns a 1 in bit 5 when the link partner has advertised 10BASE-T half-duplex capability; otherwise, it returns a 0.

Protocol Selector Field

Bits [4:0] return the value of the link partner's advertised Protocol Selector field.

1000BASE-T/100BASE-TX/10BASE-T AUTO-NEGOTIATION EXPANSION REGISTER (ADDRESS 06H)

Table 483: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Reserved	R/W	Write as 0, ignore on read	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Reserved	R/W	Write as 0, ignore on read	0
8	Reserved	R/W	Write as 0, ignore on read	0
7	Reserved	R/W	Write as 0, ignore on read	0
6	Reserved	R/W	Write as 0, ignore on read	0
5	Reserved	R/W	Write as 0, ignore on read	0
4	Parallel Detection Fault	RO	1 = Parallel link fault detected	0
		LH	0 = Parallel link fault not detected	
3	Link Partner Next Page Ability	RO	1 = Link partner has Next Page capability 0 = Link partner does not have Next Page capability	0
2	Next Page Capable	RO	1 = NetXtreme II is Next Page capable 0 = NetXtreme II is not Next Page capable	1
1	Page Received	RO	1 = New page has been received from link partner 0 = New page has not been received	0
0	Link Partner Auto-Negotiation Ability	RO	1 = Link partner has auto-negotiation capability 0 = Link partner does not have auto-negotiation	0

Parallel Detection Fault

When a parallel detection fault has occurred in the auto-negotiation state machine, bit 4 returns a 1. When a parallel detection fault occurs, this bit is latched at 1 and remains so until the register is read. If a parallel detection fault has not occurred since the last time it was read, this bit returns a 0.

Link Partner Next Page Ability

The NetXtreme II returns a 1 in bit 3 when the link partner needs to transmit Next Page information; otherwise, it returns a 0. This bit is a copy of bit 15 of the 1000BASE-T/100BASE-TX/10BASE-T Link Partner Ability register.

Next Page Capable

When bit 2 is read, the NetXtreme II supports Next Page capability and returns a 1.

Page Received

The NetXtreme II returns a 1 in bit 1 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

Link Partner Auto-Negotiation Ability

When the link partner is known to have auto-negotiation capability, the NetXtreme II returns a 1 in bit 0. Before any auto-negotiation information is exchanged (or if the link partner does not comply with IEEE Auto-Negotiation), the bit returns a 0.

1000BASE-T/100BASE-TX/10BASE-T NEXT PAGE TRANSMIT REGISTER (ADDRESS 07H)

Table 484: 1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h)

Bit	Name	R/W	Description	Default
15	Next Page	R/W	1 = Additional next pages follow 0 = Sending last Next Page	0
14	Reserved	RO	Ignore on read	0
13	Message Page	R/W	1 = Formatted page 0 = Unformatted page	1
12	Acknowledge2	R/W	1 = Complies with message (Not used during 1000BASE-T next pages) 0 = Cannot comply with message	0
11	Toggle	RO	Toggles between exchanges of differing next pages	0
10	Message/Unformatted Code Field	R/W	Next page message code or unformatted data	0
9		R/W		0
8		R/W		0
7		R/W		0
6		R/W		0
5		R/W		0
4		R/W		0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		1

Next Page

Bit 15 must be set = 1 to indicate that more Next Pages are to be sent. This bit must be cleared to indicate that this is the last Next Page to be transmitted. When this bit is read, it returns the last value written.

Message Page

Bit 13 must be set = 1 to indicate that a formatted message page is being sent. This bit must be cleared to indicate that an unformatted page is being sent. When this bit is read, it returns the last value written.

Acknowledge2

When this bit is set = 1, the NetXtreme II indicates that it can comply with the Next Page request. When this bit is cleared, the NetXtreme II indicates that it cannot comply with the Next Page request. When this bit is read, it returns the last value written.

Toggle

This bit toggles between different Next Page exchanges to ensure a functional synchronization to the link partner.

Message/Unformatted Code Field

These 11 bits make up the message code defined by IEEE 802.3, Clause 28, Annex C, when sending formatted pages. When sending unformatted Next Pages, these 11 bits contain an arbitrary data value.

1000BASE-T/100BASE-TX/10BASE-T LINK PARTNER RECEIVED NEXT PAGE REGISTER (ADDRESS 08H)

**Table 485: 1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register
(Address 08h)**

Bit	Name	R/W	Description	Default
15	Next Page	RO	1 = Additional next pages follow 0 = Sending last Next Page	0
14	Acknowledge	RO	1 = Acknowledge 0 = No acknowledge	0
13	Message Page	RO	1 = Formatted page 0 = Unformatted page	0
12	Acknowledge2	RO	1 = Complies with message (Not used during 1000BASE-T next pages) 0 = Cannot comply with message	0
11	Toggle	RO	Toggles between exchanges of different next pages	0
10	Message Code field	RO	Next Page message code or unformatted data	0
9		RO		0
8		RO		0
7		RO		0
6		RO		0
5		RO		0
4		RO		0
3		RO		0
2		RO		0
1		RO		0
0		RO		0

Next Page

When the link partner has indicated that more Next Pages are to be sent, bit 15 returns a 1. This bit returns a 0 when the link partner has indicated that this is the last Next Page to be transmitted.

Acknowledge

Bit 14 returns a 1 to indicate that the link partner has received and acknowledged a Next Page. The bit returns a 0 until the link partner has acknowledged the page.

Message Page

Bit 13 returns a 1 to indicate that the link partner has sent a formatted message page. This bit returns a 0 when the link partner has sent an unformatted page.

Acknowledge2

When the link partner has indicated that it can comply with the Next Page request, bit 12 returns a 1. When the link partner has indicated that it cannot comply with the Next Page request, this bit returns a 0.

Toggle

To ensure a functional synchronization to the NetXtreme II transceiver, the link partner toggles this bit between different Next Page exchanges.

Message Code Field

These 11 bits make up the message code defined by IEEE 802.3, Clause 28, Annex C, when the link partner has sent a formatted page. When the link partner has sent unformatted next pages, these 11 bits contain an arbitrary data value.

1000BASE-T CONTROL REGISTER (ADDRESS 09H)

Table 486: 1000BASE-T Control Register (Address 09h)

Bit	Name	R/W	Description	Default
15	Test Mode	R/W	1 X X = Test mode 4—Transmitter distortion test 0 1 1 = Test mode 3—Slave transmit jitter test 0 1 0 = Test mode 2—Master transmit jitter test 0 0 1 = Test mode 1—Transmit waveform test 0 0 0 = Normal operation	0
14		R/W		0
13		R/W		0
12	Master/Slave Configuration Enable	R/W	1 = Enable master/slave manual configuration value 0 = Automatic master/slave configuration	0
11	Master/Slave Configuration Value	R/W	1 = Configure PHY as master 0 = Configure PHY as slave	0
10	Repeater/DTE	R/W	1 = Repeater/switch device port 0 = DTE device	0
9	Advertise 1000BASE-T Full-Duplex Capability	R/W	1 = Advertise 1000BASE-T full-duplex capability 0 = Advertise no 1000BASE-T full-duplex capability	1
8	Advertise 1000BASE-T Half-Duplex Capability	R/W	1 = Advertise 1000BASE-T half-duplex capability 0 = Advertise no 1000BASE-T half-duplex capability	1
7	Reserved	RO	Ignore on read	0
6	Reserved	RO	Ignore on read	0
5	Reserved	RO	Ignore on read	0
4	Reserved	RO	Ignore on read	0
3	Reserved	RO	Ignore on read	0
2	Reserved	RO	Ignore on read	0
1	Reserved	RO	Ignore on read	0
0	Reserved	RO	Ignore on read	0

Test Mode

The NetXtreme II can be placed in 1 of 4 transmit test modes by writing bits [15:13] of the 1000BASE-T Control register. The transmit test modes are defined in IEEE 802.3ab. When read, these bits return the last value written. For test modes 1, 2 and 4 the PHY must have auto-negotiation disabled, forced to 1000BASE-T mode and Auto-MDIX disabled.

- Disable auto-neg and Force to 1000BASE-T mode (write to register 00h = 0x0040)
- Disable Auto-MDIX (write to register 18h, shadow value 111, bit 9 = 0)
- Enter test mode s (write to register 09h, bits[15:13] = the test mode you want)

Master/Slave Configuration Enable

When bit 12 is set = 1, the NetXtreme II master/slave mode is configured using the manual master/slave configuration value. When the bit is cleared, the master/slave mode is configured using the automatic resolution function. This bit returns a 1 when manual master/slave configuration is enabled; otherwise, it returns a 0.

Master/Slave Configuration Value

When bit 12 is set = 1, bit 11 determines the NetXtreme II master/slave mode of operation. When bit 11 is set, the NetXtreme II is configured as the master. When bit 11 is cleared, the NetXtreme II is configured as the slave. When read, this bit returns the last value written.

Repeater/DTE

When bit 10 is set = 1, the NetXtreme II advertises that it is a repeater or switch device port. When the bit is cleared, the NetXtreme II advertises that it is a DTE port. The advertised value is used in the automatic master/slave configuration resolution. The link partner which advertises repeater mode is configured to master if the opposing link partner advertises data terminal equipment (DTE); otherwise, this bit has no effect. This bit returns a 1 when advertising repeater/switch mode; otherwise, it returns a 0.

Advertise 1000BASE-T Full-Duplex Capability

When bit 9 is set = 1, the NetXtreme II advertises 1000BASE-T full-duplex capability. When the bit is cleared, the NetXtreme II advertises no 1000BASE-T full-duplex capability. This bit returns a 1 when advertising 1000BASE-T full-duplex capability; otherwise, it returns a 0.

Advertise 1000BASE-T Half-Duplex Capability

When bit 8 is set = 1, the NetXtreme II advertises 1000BASE-T half-duplex capability. When the bit is cleared, the NetXtreme II advertises no 1000BASE-T half-duplex capability. This bit returns a 1 when advertising 1000BASE-T half-duplex capability; otherwise, it returns a 0.

1000BASE-T STATUS REGISTER (ADDRESS 0Ah)*Table 487: 1000BASE-T Status Register (Address 0Ah)*

Bit	Name	R/W	Description	Default
15	Master/Slave Configuration Fault	RO LH	1 = Master/slave configuration fault detected 0 = No master/slave configuration fault detected	0
14	Master/Slave Configuration Resolution	RO	1 = Local transmitter is master 0 = Local transmitter is slave	0
13	Local Receiver Status	RO	1 = Local receiver OK 0 = Local receiver not OK	0
12	Remote Receiver Status	RO	1 = Remote receiver OK 0 = Remote receiver not OK	0
11	Link Partner 1000BASE-T Full-Duplex Capability	RO	1 = Link partner is 1000BASE-T full-duplex capable 0 = Link partner not 1000BASE-T full-duplex capable	0
10	Link Partner 1000BASE-T Half-Duplex Capability	RO	1 = Link partner is 1000BASE-T half-duplex capable 0 = Link partner not 1000BASE-T half-duplex capable	0
9	Reserved	RO	Ignore on read	0
8	Reserved	RO	Ignore on read	0
7	Idle Error Count	RO CR	Number of idle errors since last read	0
6		RO CR		0
5		RO CR		0
4		RO CR		0
3		RO CR		0
2		RO CR		0
1		RO CR		0
0		RO CR		0



Note: As indicated by bit 5 of the MII Status register, the values contained in bits 14, 11, and 10 of the 1000BASE-T Status register are guaranteed to be valid only after auto-negotiation has successfully completed.

Master/Slave Configuration Fault

When a master/slave configuration fault has occurred during auto-negotiation, the NetXtreme II returns a 1 in bit 15. When a configuration fault occurs, the bit is latched at 1 and remains so until either the register is read, auto-negotiation is restarted by writing bit 9 in the MII Control register, or auto-negotiation completes successfully with no master/slave configuration fault.

Master/Slave Configuration Resolution

When the NetXtreme II transceiver has been configured as the master, it returns a 1 in bit 14. When the NetXtreme II transceiver has been configured as the slave, it returns a 0.

Local Receiver Status

The NetXtreme II transceiver returns a 1 in bit 13 when the local receiver status is OK; otherwise, it returns a 0.

Remote Receiver Status

The NetXtreme II returns a 1 in bit 12 when the remote receiver status is OK; otherwise, it returns a 0.

1000BASE-T Full-Duplex Capability

The NetXtreme II returns a 1 in bit 11 when the link partner has advertised 1000BASE-T full-duplex capability; otherwise, it returns a 0.

1000BASE-T Half-Duplex Capability

The NetXtreme II returns a 1 in bit 10 when the link partner has advertised 1000BASE-T half-duplex capability; otherwise, it returns a 0.

Idle Error Count

The NetXtreme II counts the number of idle errors received while the local receiver status is OK. Bits 7 through 0 return the number of idle errors counted since the last time the register was read. The counter freezes at the maximum value (FFh) to prevent overflow.

1000BASE-T/100BASE-TX/10BASE-T IEEE EXTENDED STATUS REGISTER (ADDRESS 0FH)

Table 488: 1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh)

Bit	Name	R/W	Description	Default
15	1000BASE-X Full-Duplex Capable	RO L	1 = 1000BASE-X full-duplex capable 0 = Not 1000BASE-X full-duplex capable	0
14	1000BASE-X Half-Duplex Capable	RO L	1 = 1000BASE-X half-duplex capable 0 = Not 1000BASE-X half-duplex capable	0
13	1000BASE-T Full-Duplex Capable	RO H	1 = 1000BASE-T full-duplex capable 0 = Not 1000BASE-T full-duplex capable	1
12	1000BASE-T Half-Duplex Capable	RO H	1 = 1000BASE-T half-duplex capable 0 = Not 1000BASE-T half-duplex capable	1
11	Reserved	RO	Ignore on read	0
10	Reserved	RO	Ignore on read	0
9	Reserved	RO	Ignore on read	0
8	Reserved	RO	Ignore on read	0
7	Reserved	RO	Ignore on read	0
6	Reserved	RO	Ignore on read	0
5	Reserved	RO	Ignore on read	0
4	Reserved	RO	Ignore on read	0
3	Reserved	RO	Ignore on read	0
2	Reserved	RO	Ignore on read	0
1	Reserved	RO	Ignore on read	0
0	Reserved	RO	Ignore on read	0

1000BASE-X Full-Duplex Capable

The NetXtreme II is not capable of 1000BASE-X full-duplex operation, and returns a 0 when bit 15 is read.

1000BASE-X Half-Duplex Capable

The NetXtreme II is not capable of 1000BASE-X half-duplex operation, and returns a 0 when bit 14 is read.

1000BASE-T Full-Duplex Capable

The NetXtreme II is capable of 1000BASE-T full-duplex operation, and returns a 1 when bit 13 is read.

1000BASE-T Half-Duplex Capable

The NetXtreme II is capable of 1000BASE-T half-duplex operation, and returns a 1 when bit 12 is read.

1000BASE-T/100BASE-TX/10BASE-T PHY EXTENDED CONTROL REGISTER (ADDRESS 10H)

Table 489: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h)

Bit	Name	R/W	Description	Default
15	MAC/PHY Interface Mode	R/W	1 = RTBI Interface (RGMII mode must be enabled) 0 = GMII/RGMII Interface	0
14	Disable Automatic MDI Crossover	R/W	1 = Automatic MDI crossover disabled 0 = Automatic MDI crossover enabled	0
13	Transmit Disable	R/W	1 = Transmitter outputs disabled 0 = Normal operation	0
12	Interrupt Disable	R/W	1 = Interrupt status output disabled 0 = Interrupt status output enabled	0
11	Force Interrupt	R/W	1 = Force interrupt status to active 0 = Normal operation	0
10	Bypass 4B/5B Encoder/Decoder (100BASE-T)	R/W	1 = Transmit and receive 5B codes over MII pins 0 = Normal MII	0
9	Bypass Scrambler/Descrambler (100BASE-T)	R/W	1 = Scrambler and descrambler disabled 0 = Scrambler and descrambler enabled	0
8	Bypass MLT3 Encoder/Decoder (100BASE-T)	R/W	1 = Bypass NRZI/MLT3 encoder and decoder 0 = Normal operation	0
7	Bypass Receive Symbol Alignment (100BASE-T)	R/W	1 = 5B receive symbols not aligned 0 = Receive symbols aligned to 5B boundaries	0
6	Reset Scrambler (100BASE-T)	R/W SC	1 = Reset scrambler to initial state 0 = Normal scrambler operation	0
5	Enable LED Traffic mode	R/W	1 = LED traffic mode enabled 0 = LED traffic mode disabled	0
4	Force LEDs On	R/W	1 = Force all LEDs into on state 0 = Normal LED operation	0
3	Force LEDs Off	R/W	1 = Force all LEDs into off state 0 = Normal LED operation	0
2	Reserved	R/W	Write as 0, ignore on read	0
1	Reserved	R/W	Write as 0, ignore on read	0
0	1000Mbps PCS Transmit FIFO Elasticity (Jumbo Packets)	R/W	1 = High latency (Up to 10K Byte Packets) 0 = Low latency (Up to 4.5K Byte Packets)	0

MAC/PHY Interface Mode

Setting bit 15 = 1 and having the NetXtreme II in RGMII mode will put the NetXtreme II in RTBI mode. Setting this bit = 0 puts the part into GMII or RGMII mode depending on the setting of INTF_SEL[3:0] pins.

Disable Automatic MDI Crossover

The automatic MDI crossover function can be disabled by setting bit 14 = 1. When the bit is cleared, the NetXtreme II performs the automatic MDI crossover function.

Transmit Disable

The transmitter can be disabled by setting bit 13 = 1. The transmitter outputs (TRD \pm [3:0]) are forced into a high impedance state.

Interrupt Disable

When this bit is set = 1, the INTR LED pin is forced to its inactive state.

Force Interrupt

When this bit is set = 1, the INTR LED pin is forced to its active state.

Bypass 4B/5B Encoder/Decoder (100BASE-T)

The 100BASE-TX 4B/5B encoder/decoder can be bypassed by setting bit 10 = 1. The transmitter sends 5B codes from the TX_ER and TXD[3:0] pins directly to the scrambler. TX_EN is ignored and frame encapsulation (insertion of J/K and T/R codes) is not performed. The receiver places descrambled and aligned 5B codes onto the RX_ER and RXD[3:0] pins. CRS is still asserted when a valid frame is received.

Bypass Scrambler/Descrambler (100BASE-T)

The 100BASE-TX stream cipher function can be disabled by setting bit 9 = 1. The stream cipher function can be re-enabled by writing a 0 to this bit.

Bypass MLT3 Encoder/Decoder (100BASE-T)

The 100BASE-TX MLT3 encoder and decoder can be bypassed by setting bit 8 = 1. NRZ data is transmitted and received on the cable. The MLT3 encoder can be re-enabled by clearing this bit.

Bypass Receive Symbol Alignment (100BASE-T)

100BASE-TX receive symbol alignment can be bypassed by setting bit 7 = 1. When used in conjunction with the bypass 4B/5B encoder/decoder bit, unaligned 5B codes are placed directly on the RX_ER and RXD[3:0] pins.

Reset Scrambler (100BASE-T)

When bit 6 is set = 1, the NetXtreme II resets the scrambler to an all 1 state. This bit is self-clearing, and always returns 0 when read.

Enable LED Traffic Mode

When bit 5 is set = 1, the NetXtreme II enables the LED traffic mode for ACTIVITY, RCVLED and XMITLED. When the bit is cleared, the NetXtreme II disables the LED traffic mode.

Force LEDs On

When bit 4 is set = 1, the NetXtreme II forces all LEDs into the ON state. When the bit is cleared, the NetXtreme II resets all LEDs to normal operation.

Force LEDs Off

When bit 3 is set = 1, the NetXtreme II forces all LEDs into the OFF state. When the bit is cleared, the NetXtreme II resets all LEDs to normal operation.

FIFO Elasticity (1000BASE-T/1000BASE-X)

When bit 0 is set = 1, the NetXtreme II sets the 1000BASE-T/1000BASE-X FIFO elasticity to high latency. In this mode, the NetXtreme II can transmit packets up to 10 KB in length. When this bit is cleared, the 1000BASE-T/1000BASE-X FIFO elasticity is set to low latency. In this mode, the NetXtreme II can transmit packets up to 4.5 KB in length. Setting this bit to 1 adds 16 nanoseconds to the 1000BASE-T/1000BASE-X transmit latency.

1000BASE-T/100BASE-TX/10BASE-T PHY EXTENDED STATUS REGISTER (ADDRESS 11H)

Table 490: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h)

Bit	Name	R/W	Description	Default
15	Auto-Negotiation Base Page Selector Field Mismatch	RO LH	1 = Link Partner Base Page Selector field mismatched Advertised Selector field since last read 0 = No mismatch detected since last read	0
14	Reserved	RO	Ignore on read	0
13	MDI Crossover State	RO	1 = Crossover MDI mode 0 = Normal MDI mode	0
12	Interrupt Status	RO	1 = Unmasked interrupt currently active 0 = Interrupt cleared	0
11	Remote Receiver Status	RO LL	1 = Remote receiver OK 0 = Remote receiver not OK since last read	0
10	Local Receiver Status	RO LL	1 = Local receiver OK 0 = Local receiver not OK since last read	0
9	Locked	RO	1 = Descrambler locked 0 = Descrambler unlocked	0
8	Link Status	RO	1 = Link pass 0 = Link fail	0
7	CRC Error Detected	RO LH	1 = CRC error detected 0 = No CRC error since last read	0
6	Carrier Extension Error Detected	RO LH	1 = Carrier extension error detected since last read 0 = No carrier extension error since last read	0
5	Bad SSD Detected (False Carrier)	RO LH	1 = Bad SSD error detected since last read 0 = No bad SSD error since last read	0
4	Bad ESD Detected (Premature End)	RO LH	1 = Bad ESD error detected since last read 0 = No bad ESD error since last read	0

Table 490: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h) (Cont.)

Bit	Name	R/W	Description	Default
3	Receive Error Detected	RO	1 = Receive error detected since last read	0
			LH 0 = No receive error since last read	
2	Transmit Error Detected	RO	1 = Transmit error code received since last read	0
			LH 0 = No transmit error code received since last read	
1	Lock Error Detected	RO	1 = Lock error detected since last read	0
			LH 0 = No lock error since last read	
0	MLT3 Code Error Detected	RO	1 = MLT3 code error detected since last read	0
			LH 0 = No MLT3 code error since last read	

Auto-Negotiation Base Page Selector Field Mismatch

When this bit is set = 1, the auto-negotiation base page selector did not match the Advertised Selector field since the previous read. When this bit reads back a 0, there is no mismatched Page Selector field and Advertised Selector field.

MDI Crossover State

When the NetXtreme II is automatically switching the transmit and receive pairs to communicate with a remote device, the NetXtreme II returns a 1 in bit 13. This bit returns a 0 when the NetXtreme II is in normal MDI mode.

Interrupt Status

The NetXtreme II returns a 1 in bit 12 when any unmasked interrupt is currently active; otherwise, it returns a 0.

Remote Receiver Status

When the remote receiver status is OK, the NetXtreme II returns a 1 in bit 11. When the NetXtreme II detects that the remote receiver is not OK, this bit is latched at 0 and remains so until the bit is read and the remote receiver status is OK.

Local Receiver Status

When the local receiver status is OK, the NetXtreme II returns a 1 in bit 10. When the NetXtreme II detects that the local receiver is not OK, this bit is latched at 0 and remains so until the bit is read and the remote receiver status is OK.

Locked

The NetXtreme II returns a 1 in bit 9 when the descrambler is locked to the incoming data stream; otherwise, it returns a 0.

Link Status

The NetXtreme II returns a 1 in bit 8 when the device has established a link; otherwise, it returns a 0.

CRC Error Detected

The NetXtreme II returns a 1 in bit 7 if a CRC error has been detected since the last time this register was read; otherwise, it returns a 0.

Carrier Extension Error Detected

The NetXtreme II returns a 1 in bit 6 if a carrier extension error has been detected since the last time this register was read; otherwise, it returns a 0.

Bad SSD Detected (False Carrier)

The NetXtreme II returns a 1 in bit 5 if a bad start-of-stream error has been detected since the last time this register was read; otherwise, it returns a 0.

Bad ESD Detected (Premature End)

The NetXtreme II returns a 1 in bit 4 if a bad end-of-stream error has been detected since the last time this register was read; otherwise, it returns a 0.

Receive Error Detected

The NetXtreme II returns a 1 in bit 3 if a packet was received with an invalid code since the last time this register was read; otherwise, it returns a 0.

Transmit Error Detected

The NetXtreme II returns a 1 in bit 2 if a packet was received with a transmit error code since the last time this register was read; otherwise, it returns a 0.

Lock Error Detected

The NetXtreme II returns a 1 in bit 1 if the descrambler has lost lock since the last time this register was read; otherwise, it returns a 0.

MLT3 Code Error Detected

The NetXtreme II returns a 1 in bit 0 if an MLT3 coding error has been detected in the receive data stream since the last time this register was read; otherwise, it returns a 0.

1000BASE-T/100BASE-TX/10BASE-T RECEIVE ERROR COUNTER REGISTER (ADDRESS 12H)

Table 491: 1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h)^a

Bit	Name	R/W	Description	Default
15:0	Receive Error Counter	R/W CR	Number of non-collision packets with receive errors since last read	0000h

a. Bits 15:0 of this register become the 1000BASE-T, 100BASE-TX, 10BASE-T Receive Error Counter when register 1Ch, shadow value 11011, bit 9 = 0.

Copper Receive Error Counter

When bit 9 = 0 in register 1Ch, shadow value 11011, this counter increments each time the NetXtreme II receives a 1000BASE-T, 100BASE-TX, or 10BASE-T non collision packet containing at least 1 receive error. This counter freezes at the maximum value of FFFFh. The counter automatically clears when read.

1000BASE-X RECEIVE ERROR COUNTER REGISTER (ADDRESS 12H)

Table 492: 1000BASE-X Receive Error Counter Register (Address 12h)^a

Bit	Name	R/W	Description	Default
15:0	Receive Error Counter	R/W CR	Number of non-collision packets with receive errors since last read	0000h

- a. Bits 15:0 of this register become the 1000BASE-X Receive Error Counter when register 1Ch, shadow value 11011, bit 9 = 1.

Fiber Receive Error Counter

When bit 9 = 1 in register 1Ch, shadow value 11011, this counter increments each time the NetXtreme II receives a 1000BASE-X non collision packet containing at least 1 receive error. This counter freezes at the maximum value of FFFFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T FALSE CARRIER SENSE COUNTER REGISTER (ADDRESS 13H)

Table 493: 1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h)^a

Bit	Name	R/W	Description	Default
15:8	Reserved	R/W	Write as 00h, ignore on read	00h
7:0	False Carrier Sense Counter	R/W CR	Number of false carrier sense events since last read	00h

- a. Bits 7:0 of this register become the 1000BASE-T/100BASE-TX/10BASE-T Carrier Sense Counter when register 1Ch, shadow 11011, bit 9 = 1 and register 1Eh, bit 14 = 0.

Copper False Carrier Sense Counter

When bit 9 = 0 in register 1Ch, shadow value 11011 and bit 14 = 0 in register 1Eh, the False Carrier Sense Counter increments each time the NetXtreme II detects a 1000BASE-T, 100BASE-TX, or 10BASE-T false carrier sense on the receive input. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-X FALSE CARRIER SENSE COUNTER REGISTER (ADDRESS 13H)

Table 494: 1000BASE-X False Carrier Sense Counter Register (Address 13h)^a

Bit	Name	R/W	Description	Default
15:8	Reserved	R/W	Write as 00h, ignore on read	00h
7:0	False Carrier Sense Counter	R/W CR	Number of false carrier sense events since last read	00h

a. Bits 7:0 of this register become the 1000BASE-X False Carrier Sense Counter when register 1Ch, shadow 11011, bit 9 = 1 and register 1Eh, bit 14 = 0.

Fiber False Carrier Sense Counter

When bit 9 = 1 in register 1Ch, shadow value 11011 and bit 14 = 0 in register 1Eh, the False Carrier Sense Counter increments each time the NetXtreme II detects a 1000BASE-X false carrier sense on the receive input. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T PACKETS RECEIVED WITH TRANSMIT ERROR CODES COUNTER REGISTER (ADDRESS 13H)

Table 495: 1000BASE-T/100BASE-TX/10BASE-T Transmit Error Code Counter Register (Address 13h)^a

Bit	Name	R/W	Description	Default
15:8	Reserved	R/W	Write as 00h, ignore on read	00h
7:0	Transmit Error Code Counter	R/W CR	Number of packets received with transmit error codes since last read	00h

a. Bits 7:0 of this register become the 1000BASE-T/100BASE-TX/10BASE-T Packets Received with Transmit Error Codes Counter when register 1Ch, shadow 11011, bit 9 = 0 and register 1Eh, bit 14 = 1.

Packets Received with Transmit Error Codes Counter

When bit 9 = 0 in register 1Ch, shadow value 11011, and when bit 14 = 1 in register 1Eh, Packets Receive with Transmit Error Codes Counter increments each time the NetXtreme II detects a 1000BASE-T, 100BASE-TX, or 10BASE-T packet with a transmit error code violation. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-X PACKETS RECEIVED WITH TRANSMIT ERROR CODES COUNTER REGISTER (ADDRESS 13H)

Table 496: 1000BASE-X Transmit Error Code Counter Register (Address 13h)^a

Bit	Name	R/W	Description	Default
15:8	Reserved	R/W	Write as 00h, ignore on read	00h
7:0	Transmit Error Code Counter	R/W CR	Number of packets received with transmit error codes since last read	00h

a. Bits 7:0 of this register become the 1000BASE-T/100BASE-TX/10BASE-T Packets Received with Transmit Error Codes Counter when register 1Ch, shadow 11011, bit 9 = 1 and register 1Eh, bit 14 = 1.

Packets Received with Transmit Error Codes Counter

When bit 9 = 0 in register 1Ch, shadow value 11011, and when bit 14 = 1 in register 1Eh, the Packets Receive with Transmit Error Codes Counter increments each time the NetXtreme II detects a 1000BASE-X packet with a transmit error code violation. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T RECEIVER NOT_OK COUNTER REGISTER (ADDRESS 14H)

Table 497: 1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h)^a

Bit	Name	R/W	Description	Default
15:8	Local Receiver NOT_OK Counter	R/W CR	Number of times local receiver was NOT_OK since last read	00h
7:0	Remote Receiver NOT_OK Counter	R/W CR	Number of times NetXtreme II detected that the remote receiver was NOT_OK since last read	00h

a. Bits 15:0 of this register become the 1000BASE-T, 100BASE-TX, or 10BASE-T Receiver NOT_OK Counter when register 1Ch, shadow 11011, bit 9 = 0 and register 1Eh bit 15 = 0.

Copper Local Receiver NOT_OK Counter

When bit 9 = 0 in register 1Ch, shadow value 11011 and bit 15 = 0 in register 1Eh, this counter increments each time the 1000BASE-T, 100BASE-TX, or 10BASE-T local receiver enters the NOT_OK state. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

Copper Remote Receiver NOT_OK Counter

When bit 9 = 0 in register 1Ch, shadow value 11011 and bit 15 = 0 in register 1Eh, this counter increments each time the 1000BASE-T, 100BASE-TX, or 10BASE-T remote receiver enters the NOT_OK state. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-X RECEIVER NOT_OK COUNTER REGISTER (ADDRESS 14H)

Table 498: 1000BASE-X Receiver NOT_OK Counter Register (Address 14h)^a

Bit	Name	R/W	Description	Default
15:8	Local Receiver NOT_OK Counter	R/W CR	Number of times local receiver was NOT_OK since last read	00h
7:0	Remote Receiver NOT_OK Counter	R/W CR	Number of times NetXtreme II detected that the remote receiver was NOT_OK since last read	00h

a. Bits 15:0 of this register become the 1000BASE-X Receiver NOT_OK Counter when register 1Ch, shadow 11011, bit 9 = 1 and register 1Eh bit 15 = 0.

Fiber Local Receiver NOT_OK Counter

When bit 9 = 1 in register 1Ch, shadow value 11011 and bit 15 = 0 in register 1Eh, this counter increments each time the 1000BASE-X local receiver enters the NOT_OK state. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

Fiber Remote Receiver NOT_OK Counter

When bit 9 = 1 in register 1Ch, shadow value 11011 and bit 15 = 0 in register 1Eh, this counter increments each time the 1000BASE-X remote receiver enters the NOT_OK state. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T RECEIVE CRC COUNTER REGISTER (ADDRESS 14H)

Table 499: 1000BASE-T/100BASE-TX/10BASE-T CRC Counter Register (Address 14h)^a

Bit	Name	R/W	Description	Default
15:0	Receive CRC Counter	R/W CR	Number of times Receive CRC errors were detected	00h

a. Bits 15:0 of this register become the 1000BASE-T, 100BASE-TX, or 10BASE-T Receiver NOT_OK Counter when register 1Ch, shadow 11011, bit 9 = 0 and register 1Eh bit 15 = 1.

Copper CRC Counter

When bit 9 = 0 in register 1Ch, shadow value 11011 and bit 15 = 1 in register 1Eh, this counter increments each time the 1000BASE-T, 100BASE-TX, or 10BASE-T detects a receive CRC error. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-X RECEIVE CRC COUNTER REGISTER (ADDRESS 14H)

Table 500: 1000BASE-X Receive CRC Counter Register (Address 14h)^a

Bit	Name	R/W	Description	Default
15:0	Receive CRC Counter	R/W CR	Number of times Receive CRC errors were detected	00h

- a. Bits 15:0 of this register become the 1000BASE-X Receiver NOT_OK Counter when register 1Ch, shadow 11011, bit 9 = 1 and register 1Eh bit 15 = 1.

Fiber CRC Counter

When bit 9 = 1 in register 1Ch, shadow value 11011 and bit 15 = 1 in register 1Eh, this counter increments each time the 1000BASE-X detects a receive CRC error. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T EXPANSION ACCESS REGISTER (ADDRESS 17H)

Table 501: 1000BASE-T/100BASE-TX/10BASE-T Expansion Access Register (Address 17h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Expansion Register Select	R/W	1111 = Expansion register selected 0000 = Expansion register not selected	0
10		R/W	0000 = Expansion register not selected	0
9		R/W		0
8		R/W	All Others = Reserved (Do not use)	0
7	Expansion Register Accessed	R/W	Sets the Expansion register number accessed when read/write to register 15h.	0
6		R/W		0
5		R/W		0
4		R/W		0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		0

Expansion Register Select

Setting bits[11:8] to 1111 enables the reading from and writing to the Expansion registers in conjunction with register 15h. These bits should be cleared after the Expansion registers are accessed or when the Expansion registers are not being accessed. See “[Expansion Registers](#)” on page 833 for Expansion register detail.

Expansion Register Accessed

Bits[7:0] which Expansion register is accessed. The Expansion registers can be accessed through register 15h when bits[11:8] of this register are set to 1111. The available expansion registers are listed in [Table 570](#).

Table 502: Expansion Register Select Values

Expansion Register	Register Name
00h	"Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)" on page 725
01h	"Expansion Register 01h: Expansion Interrupt Status Register (Address 15h)" on page 725
04h	"Expansion Register 04h: Multicolor LED Selector Register (Address 15h)" on page 834
05h	"Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)" on page 836
06h	"Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)" on page 837

1000BASE-T/100BASE-TX/10BASE-T AUXILIARY CONTROL SHADOW VALUE ACCESS REGISTER (ADDRESS 18H)

Available 18h registers are listed in [Table 571](#):

Table 503: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Shadow Values Access Register (Address 18h)

Shadow Value	Register Name
000	"1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow 0h)" on page 771
001	"10BASE-T Register (Address 18h, Shadow 1h)" on page 773
010	"1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow 2h)" on page 775
100	"1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow 4h)" on page 777
111	"1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow 7h)" on page 779

Read from register 18h, shadow value zzz:

Table 504: Reading Register 18h

Register Reads/Writes	Description
Write register 18h, bits[2:0] = 111	This selects the Misc Control register, shadow value 111. All reads must be done through the Misc Control register.
Bit 15 = 0	This allows only bits[14:12] and bits[2:1] to be written.
Bits[14:12] = zzz	This selects shadow value register zzz to be read.
Bits[11: 3] = <don't care>	When bit 15 = 0, these bits will be ignored.
Bits[2:0] = 111	This sets the Shadow Register Select to 111 (Misc Control register).



Table 504: Reading Register 18h

Register Reads/Writes	Description
Read register 18h	Data read back is the value from shadow register zzz.

Write to Register 18h, shadow value yyy:

Table 505: Writing Register 18h

Register Writes	Description
Set Bits[15:3] = Preferred write values	Bits[15:3] contain the desired bits to be written to.
Set Bits[2:0] = yyy	This enables shadow value register yyy to be written. For shadow value 111, bit 15 must also be written.

1000BASE-T/100BASE-TX/10BASE-T AUXILIARY CONTROL REGISTER (ADDRESS 18H, SHADOW VALUE 000)

Table 506: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow Value 000)

Bit	Name	R/W	Description	Default
15	External Loopback	R/W	1 = External Loopback enabled 0 = Normal operation	0
14	Receive Extended Packet Length	R/W	1 = Allow reception of extended length packets 0 = Allow reception of normal length Ethernet packets only	INTF_SEL[3:2]=11
13	Edge Rate Control (1000BASE-T)	R/W	00 = 4.0 ns	0
12		R/W	01 = 5.0 ns 10 = 3.0 ns 11 = 0.0 ns	ER pin
11	Reserved	R/W	Write as 0, ignore on read	0
10	Reserved	R/W	Write as 1, ignore on read	1
9	Reserved	R/W	Write as 0, ignore on read	0
8	Reserved	R/W	Write as 0, ignore on read	0
7	Reserved	R/W	Write as 0, ignore on read	0
6	Reserved	R/W	Write as 0, ignore on read	0
5	Edge Rate Control (100BASE-TX)	R/W	00 = 4.0 ns	0
4		R/W	01 = 5.0 ns 10 = 3.0 ns 11 = 0.0 ns	ER pin
3	Reserved	R/W	Write as 0, ignore on read	0

**Table 506: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register
(Address 18h, Shadow Value 000) (Cont.)**

Bit	Name	R/W	Description	Default
2	Shadow Register Select	R/W	000 = Auxiliary Control register	0
1		R/W	001 = 10BASE-T register	0
0		R/W	010 = Power/MII Control register 011 = Reserved 100 = Misc Test register 101 = Reserved 110 = Reserved 111 = Misc Control register	0

External Loopback

When bit 15 = 1, external loopback operation is enabled. When the bit is cleared, normal operation resumes.

Receive Extended Packet Length

When bit 14 = 1, the NetXtreme II receives 10BASE-T and 100BASE-TX packets up to 18 KB in length when in GMII, MII, or RGMII mode and 10 KB in length when in SGMII or GBIC mode. When the bit is cleared, the NetXtreme II only receives packets up to 4.5 KB in length. When in GBIC mode, extended packet length is automatically enabled.

Edge Rate Control (1000BASE-T)

Bits[13:12] control the edge rate of the 1000BASE-T transmit DAC output waveform.

Edge Rate Control (100BASE-TX)

Bits[5:4] control the edge rate of the 100BASE-TX transmit DAC output waveform.

Shadow Register Select

The Auxiliary Control register provides access to eight registers using a shadow technique. The lower 3 bits define which set of 13 upper bits are used in accordance with [Table 574 on page 771](#), defined under bits[2:0]. See the note on [“Auxiliary Control Shadow Value Access Register \(Address 18h\)” on page 770](#) describing reading from and writing to register 18h.

The register set shown above is that for normal operation, obtained when the lower 3 bits are 000.

10BASE-T REGISTER (ADDRESS 18H, SHADOW VALUE 001)*Table 507: 10BASE-T Register (Address 18h, Shadow Value 001)*

Bit	Name	R/W	Description	Default
15	Manchester Code Error	RO LH	1 = Manchester code error (10BASE-T) 0 = No Manchester code error	0
14	EOF Error	RO LH	1 = EOF error detected (10BASE-T) 0 = No EOF error detected	0
13	Polarity Error	RO	1 = Channel polarity inverted 0 = Channel polarity correct	0
12	Block RX_DV Extension (IPG)	R/W	1 = Block RX_DV for four additional RXC cycles for IPG 0 = Normal operation	0
11	10BASE-T TXC Invert Mode	R/W	1 = Invert TXC output 0 = Normal operation	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Jabber Disable	R/W	1 = Jabber function disabled 0 = Jabber function enabled	0
8	Reserved	R/W	Write as 0, ignore on read	0
7	Reserved	R/W	Write as 0, ignore on read	0
6	10BASE-T Echo Mode	R/W	1 = Echo transmit data to receive data 0 = Normal operation	0
5	SQE Enable Mode	R/W	1 = Enable SQE 0 = Disable SQE	0
4	10BASE-T No Dribble	R/W	1 = Correct 10BASE-T dribble nibble 0 = Normal operation	0
3	Reserved	R/W	Write as 0, ignore on read	0
2	Shadow Register Select	R/W	000 = Auxiliary Control register 001 = 10BASE-T register	0
1		R/W	010 = Power/MII Control register 011 = Reserved	0
0		R/W	100 = Misc Test register 101 = Reserved 110 = Reserved 111 = Misc Control register	1

Manchester Code Error

When a Manchester code violation is received, bit 15 returns a 1. This bit is valid only during 10BASE-T operation.

EOF Error

When the end-of-frame (EOF) sequence was improperly received (or not received at all), bit 14 returns a 1. This bit is valid only during 10BASE-T operation.

Polarity Error

When an analog input polarity error has been detected and corrected, bit 13 returns a 1. This bit is valid only during 10BASE-T operation.

Block RX_DV Extension (IPG)

When bit 12 of the 10BASE-T register is set, blocking of the RX_DV signal is extended for an additional 4 RXC cycles to extend the IPG.

10BASE-T TXC Invert Mode

When bit 11 of the 10BASE-T register is set, the polarity of the 10BASE-T transmit clock is inverted. Clearing this bit restores normal transmit clock polarity. This bit is valid only during 10BASE-T operation.

Jabber Disable

Setting bit 9 allows the user to disable the jabber detect function defined in the IEEE standard. When a transmission request has exceeded a maximum time limit, this function shuts off the transmitter. Clearing this bit or resetting the chip restores normal operation. Reading this bit returns the value of jabber detect disable. This bit is valid only during 10BASE-T operation.

10BASE-T Echo Mode

When bit 6 is enabled during 10BASE-T half-duplex transmit operation, the transmitted data is replicated on the receive data pins and the TXEN signal echoes on the RX_DV pin. The TXEN signal also echoes on the CRS pin and CRS deassertion directly follows the TXEN deassertion.

SQE Enable Mode

Setting bit 5 of the 10BASE-T register enables SQE mode. Clearing disables it. This bit is valid only during 10BASE-T operation.

10BASE-T No Dribble

When bit 4 is set, the PHY rounds down to the nearest nibble when dribble bits are present on the 10BASE-T input stream.

Shadow Register Select

The 10BASE-T register provides access to 8 registers using a shadow technique. The lower 3 bits written define which set of 13 upper bits are used in accordance with [Table 575 on page 773](#), defined under bits [2:0]. See the note on [“Auxiliary Control Shadow Value Access Register \(Address 18h\)” on page 770](#) describing reading from and writing to register 18h.

1000BASE-T/100BASE-TX/10BASE-T POWER/MII CONTROL REGISTER (ADDRESS 18H, SHADOW VALUE 010)

*Table 508: 1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register
(Address 18h, Shadow Value 010)*

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Reserved	R/W	Write as 0, ignore on read	0
10	Reserved	R/W	Write as 1, ignore on read	1
9	Reserved	R/W	Write as 0, ignore on read	0
8	Reserved	R/W	Write as 1, ignore on read	1
7	Reserved	R/W	Write as 0, ignore on read	0
6	Reserved	R/W	Write as 0, ignore on read	0
5	Super Isolate (Copper Only)	R/W	1 = Isolate mode with no link pulses transmitted 0 = Normal operation	0
4	Reserved	R/W	Write as 0, ignore on read	0
3	Reserved	R/W	Write as 0, ignore on read	0
2	Shadow Register Select	R/W	000 = Auxiliary Control register	0
1		R/W	001 = 10BASE-T register	1
0		R/W	010 = Power/MII Control register 011 = Reserved 100 = Misc Test register 101 = Reserved 110 = Reserved 111 = Misc Control register	0

Super Isolate (Copper Only)

Setting bit 5 = 1, places the NetXtreme II into the super isolate mode. The Copper Media Dependent Interface (MDI) is disabled and all the GMII/RGMII inputs are ignored and all GMII/RGMII outputs are tristated.

When in SGMII mode, the transmit and receive functions on the Copper Media Dependent Interface are disabled. The SGMII interface will be in auto-negotiation mode, link will be established with the Switch if the Switch sends back an acknowledgement to the PHY. If the Switch does not send back an acknowledgement, the PHY will remain in auto-negotiation mode. Any data received from the Switch/MAC will be ignored and no data will be sent from the PHY to the Switch/MAC. Only auto-negotiation link code words will pass between the PHY and the Switch since the PHY's Copper MDI is disabled and the PHY will be telling the Switch that there is no Copper link.

Shadow Register Select

The 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control register provides access to 8 registers using a shadow technique. The lower 3 bits written define which set of 13 upper bit are used in accordance with the table defined under bits

[2:0] above. See the note on “Auxiliary Control Shadow Value Access Register (Address 18h)” on page 770 describing reading from and writing to register 18h.

1000BASE-T/100BASE-TX/10BASE-T MISCELLANEOUS TEST REGISTER (ADDRESS 18H, SHADOW VALUE 100)

Table 509: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow Value 100)

Bit	Name	R/W	Description	Default
15	Lineside [Remote] Loopback Enable	R/W	1 = Enable lineside [remote] loopback 0 = Disable loopback	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Lineside [Remote] Loopback Tri-state	R/W	1 = Tristate the receive GMII/RGMII pins when lineside loopback is enabled 0 = Lineside [remote] loopback packets appear on MII	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Reserved	R/W	Write as 0, ignore on read	0
8	Reserved	R/W	Write as 0, ignore on read	0
7	Reserved	R/W	Write as 0, ignore on read	0
6	Reserved	R/W	Write as 0, ignore on read	0
5	Reserved	R/W	Write as 0, ignore on read	0
4	Swap RX MDIX	R/W	1 = RX and TX operate on same pair 0 = Normal operation	0
3	10BASE-T Halfout	R/W	1 = Transmit 10BASE-T at half amplitude 0 = Normal operation	0
2	Shadow Register Select	R/W	000 = Auxiliary Control register	1
1		R/W	001 = 10BASE-T register	0
0		R/W	010 = Power/MII Control register 011 = Reserved 100 = Misc Test register 101 = Reserved 110 = Reserved 111 = Misc Control register	0

Lineside [Remote] Loopback Enable

Setting bit 15 = 1 enables lineside [remote] loopback of the copper receive packet back out through the MDI transmit path.

Lineside [Remote] Loopback Tri-state

Setting bit 11 = 1 tri-states the receive GMII/RGMII pins when the device is in lineside [remote] loopback mode.

Swap RX MDIX

When bit 4 = 1, the transmitter and receiver operate on the same twisted pair. This function is for use in a test mode where the transmitter output is detected by the receiver attached to the same pair.

10BASE-T Halfout

When operating in 10BASE-T mode, setting bit 3 = 1 reduces the output of the transmitter to half of its normal amplitude. Clearing this bit restores full amplitude operation. This function is used in a test mode where an unterminated output generates a signal with twice the amplitude of a terminated output.

Shadow Register Select

The Misc. Test register provides access to 8 registers using a shadow technique. The lower 3 bits written define which set of 13 upper bits are used in accordance with [Table 577](#) under bits [2:0]. See the note on "[Auxiliary Control Shadow Value Access Register \(Address 18h\)](#)" on page [770](#) describing reading from and writing to register 18h.

T1000BASE-T/100BASE-TX/10BASE-T MISCELLANEOUS CONTROL REGISTER (ADDRESS 18H, SHADOW VALUE 111)

Table 510: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow Value 111)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W SC	1 = Write bits [14:0] 0 = Only write bits [14:12] and [2:0]	0
14	Shadow Register Read Selector	R/W	These bits are written when bit 15 is not set.	0
13		R/W	This sets the shadow value for address 18h register read.	0
12		R/W	000 = Normal operation 001 = 10BASE-T register 010 = Power Control register 011 = Reserved 100 = Misc Test register 101 = Reserved 110 = Reserved 111 = Misc Control register	0
11	Packet Counter Mode	R/W	1 = Receive packet Counter 0 = Transmit Packet Counter	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Force Auto-MDIX Mode	R/W	1 = Auto-MDIX is enabled when auto-negotiation is disabled 0 = Auto-MDIX is disabled when auto-negotiation is disabled	0
8	RGMII Timing Mode	R/W	1 = RGMII RXC delayed timing mode 0 = RGMII RXC/RXD aligned timing mode	0
7	RGMII Mode	R/W	1 = RGMII mode enable 0 = Normal GMII/MII operation	0
6	RGMII RX_DV Mode	R/W	1 = Mux RX_ER with RX_DV in RGMII mode 0 = Mux CRS with RX_DV in RGMII mode	1

**Table 510: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register
(Address 18h, Shadow Value 111) (Cont.)**

Bit	Name	R/W	Description	Default
5	RGMII In-Band Status Disable	R/W	1 = Send regular data during IPG 0 = Send Optional In-band status info in RGMII mode	1
4	Reserved	R/W	Write as 0, ignore on read	0
3	MDIO All PHY Select	R/W	1 = The PHY will accept MDIO writes to PHY Address = 00000 0 = Normal operation	0
2	Shadow Register Select	R/W	000 = Auxiliary register	1
1		R/W	001 = 10BASE-T register	1
0		R/W	010 = Power Control register 011 = Reserved 100 = Misc Test register 101 = Reserved 110 = Reserved 111 = Misc Control register	1

Write Enable

When bit 15 is set = 1 then bits [14:0] of this register can be modified. Bits [2:0] and [14:12] can always be written regardless of the state of bit 15.

Shadow Register Read Selector

Bits [14:12] are written, regardless of the value of bit 15. These bits determine the shadow value for an MII register 18h read operation. See the note on “[Auxiliary Control Shadow Value Access Register \(Address 18h\)](#)” on page 770 describing reading from and writing to register 18h.

Packet Counter Mode

Bit 11 sets the packet counter mode in Expansion register 00h. If bit 11 = 1, it counts the number of receive packets. If bit 11 = 0 it counts the number of transmit packets.

Force Auto-MDIX Mode

Setting bit 9 = 1 enables Auto-MDIX mode while auto-negotiation is disabled. The default is to disable the Auto-MDIX function when auto-negotiation is disabled.

RGMII Timing Mode

Bit 8 adjusts the RGMII (RX_DV/RXD[3:0] to RXC) timing. When this bit is cleared, the RXC to RXD[3:0] timing is aligned. When this bit is set = 1, the RXC is delayed 1.9 nanoseconds with respect to RXD[3:0]. The RXC delay does not function when in RGMII to Fiber mode or RTBI to Fiber mode.

RGMII Mode

The NetXtreme II uses the RGMII MAC interface when bit 7 is set = 1; otherwise, it uses the standard GMII/MII MAC interface. When INTF_SEL[0] = 1 this bit defaults to 1.

RGMII RX_DV Mode

If bit 6 is set = 1 in RGMII mode, RX_ER is multiplexed with RX_DV. If this bit is cleared, CRS is multiplexed with RX_DV.

RGMII IN-Band Status (Optional) Disable

When bit 5 = 1, the NetXtreme II sends out normal IPG. When bit 5 = 0, the NetXtreme II sends out In-Band Status data as shown in section 3.4.1 in the RGMII version 2.0 specification.

MDIO All PHY Select

When bit 3 = 1, a MDIO write operation with PHY Address = 00000 will be accepted in addition to the PHYs real address. By setting this bit to a 1 for all four ports a single write to PHY address 00000 will write all four ports.

Shadow Register Select

Using a shadow technique, the 1000BASE-T/100BASE-TX/10BASE-T Misc. Control register provides access to 8 registers. The lower 3 bits written define which set of 13 upper bits are used in accordance with [Table 578](#), defined under bits 2:0. See the note on "[Auxiliary Control Shadow Value Access Register \(Address 18h\)](#)" on page 770 describing reading from and writing to register 18h.



1000BASE-T/100BASE-TX/10BASE-T AUXILIARY STATUS SUMMARY REGISTER (ADDRESS 19H)

Table 511: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h)

Bit	Name	R/W	Description	Default
15	Auto-Negotiation Complete	RO	1 = Auto-negotiation complete 0 = Auto-negotiation in progress	0
14	Auto-Negotiation Complete Acknowledge	RO LH	1 = Entered auto-negotiation link good check state 0 = State not entered since last read	0
13	Auto-Negotiation Acknowledge Detect	RO LH	1 = Entered auto-negotiation acknowledge detect state 0 = State not entered since last read	0
12	Auto-Negotiation Ability Detect	RO LH	1 = Entered auto-negotiation ability detect state 0 = State not entered since last read	0
11	Auto-Negotiation Next Page Wait	RO LH	1 = Entered auto-negotiation Next Page wait state 0 = State not entered since last read	0
10	Auto-Negotiation HCD	RO	111 = 1000BASE-T full-duplex ^a	0
9	Current Operating Speed and Duplex Mode	RO	110 = 1000BASE-T half-duplex ^a	0
8		RO	101 = 100BASE-TX full-duplex ^a 100 = 100BASE-T4 011 = 100BASE-TX half-duplex ^a 010 = 10BASE-T full-duplex ^a 001 = 10BASE-T half-duplex ^a 000 = No highest common denominator or auto-negotiation not complete	0
7	Parallel Detection Fault	RO LH	1 = Parallel link fault detected 0 = Parallel link fault not detected	0
6	Remote Fault	RO	1 = Link partner has detected remote fault 0 = Link partner has not detected remote fault	0
5	Auto-Negotiation Page Received	RO LH	1 = New page has been received from link partner 0 = New page has not been received	0
4	Link Partner Auto-Negotiation Ability	RO	1 = Link partner has auto-negotiation capability 0 = Link partner does not perform auto-negotiation	0
3	Link Partner Next Page Ability	RO	1 = Link partner has Next Page capability 0 = Link partner does not have Next Page capability	0
2	Link Status	RO	1 = Link is up (link pass state) 0 = Link is down (link fail state)	0
1	Pause Resolution—Receive Direction	RO	1 = Enable pause receive 0 = Disable pause receive	0
0	Pause Resolution—Transmit Direction	RO	1 = Enable pause transmit 0 = Disable pause transmit	0

a. Indicates the negotiated HCD when Auto-Negotiation Enable = 1, or indicates the manually selected speed and duplex mode when Auto-Negotiation Enable = 0

Auto-Negotiation Complete

When auto-negotiation is complete, the NetXtreme II returns a 1 in bit 15. This bit returns a 0 while auto-negotiation is in progress.

Auto-Negotiation Complete Acknowledge

The NetXtreme II returns a 1 in bit 14 when the auto-negotiation state machine has entered the link good check state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation Acknowledge Detect

The NetXtreme II returns a 1 in bit 13 when the auto-negotiation state machine has entered the acknowledge detect state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation Ability Detect

The NetXtreme II returns a 1 in bit 12 when the auto-negotiation state machine has entered the ability detect state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation Next Page Wait

The NetXtreme II returns a 1 in bit 11 when the auto-negotiation state machine has entered the Next Page wait state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation HCD (Current Operating Speed and Duplex Mode)

Bits[10:8] report the mode of operation negotiated between the NetXtreme II and its link partner. As reported by bit 15, these bits return 000 until auto-negotiation is completed. When the auto-negotiation function has been disabled, bits[10:8] report the manually selected mode of operation. When the auto-negotiation function has been disabled, bits[10:8] report the manually selected mode of operation when Register 18h, shadow value 111, bit 9 = 0. If auto-negotiation is disabled and Register 18h, shadow value 111, bit 9 = 1, then bits [10:8] = 000.

Parallel Detection Fault

When a parallel detection fault has occurred in the auto-negotiation state machine, bit 7 returns a 1. When a parallel detection fault occurs, this bit is latched to a 1 and remains so until the next register read. This bit returns a 0 when a parallel detection fault has not occurred since the last time it was read.

Remote Fault

The NetXtreme II returns a 1 in bit 6 when the link partner has detected a remote fault; otherwise, it returns a 0.

Auto-Negotiation Page Received

The NetXtreme II returns a 1 in bit 5 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

Link Partner Auto-Negotiation Ability

The NetXtreme II returns a 1 in bit 4 of the when the link partner is known to have auto-negotiation capability. Before any auto-negotiation information is exchanged or if the link partner does not comply with IEEE Auto-Negotiation, the bit returns a 0.



Link Partner Next Page Ability

The NetXtreme II returns a 1 in bit 3 when the link partner needs to transmit Next Page information; otherwise, it returns a 0.

Link Status

The NetXtreme II returns a 1 in bit 2 when the link status is good; otherwise, it returns a 0.

Pause Resolution—Receive Direction and Transmit Direction

When auto-negotiation has completed, the NetXtreme II returns the result of the pause resolution function for full-duplex flow control on bits [1:0]. When bit 1 returns a 1, the link partner can send pause frames toward the local device. When bit 0 returns a 1, pause frames can be transmitted by the local device to the link partner. These bits are only guaranteed to be valid when bit 15 of the Auxiliary Status Summary register is 1.

1000BASE-T/100BASE-TX/10BASE-T INTERRUPT STATUS REGISTER (ADDRESS 1Ah)**Table 512: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah)**

Bit	Name	R/W	Description	Default
15	Signal Detect/Energy Detect Change	RO LH	1 = Filtered fiber signal detect change or energy detect change since last read (enabled by register 1Ch, shadow 00101, bit 5 = 1) 0 = Interrupt cleared	0
14	Illegal Pair Swap	RO LH	1 = Illegal pair swap detected 0 = Interrupt cleared	0
13	MDIX Status Change	RO LH	1 = MDIX status changed since last read 0 = Interrupt cleared	0
12	Exceeded High Counter Threshold	RO	1 = Value in one or more counters is above 32K 0 = All counters below 32K	0
11	Exceeded Low Counter Threshold	RO	1 = Value in one or more counters is above 128K 0 = All counters below 128K	0
10	Auto-Negotiation Page Received	RO LH	1 = Page received since last read 0 = Interrupt cleared	0
9	No HCD Link	RO LH	1 = Negotiated HCD, did not establish link 0 = Interrupt cleared	0
8	No HCD	RO LH	1 = Auto-negotiation returned HCD = none 0 = Interrupt cleared	0
7	Negotiated Unsupported HCD	RO LH	1 = Auto-negotiation HCD not supported by NetXtreme II 0 = Interrupt cleared	0
6	Scrambler Synchronization Error	RO LH	1 = Scrambler synchronization error occurred since last read 0 = Interrupt cleared	0
5	Remote Receiver Status Change	RO LH	1 = Remote receiver status changed since last read 0 = Interrupt cleared	0
4	Local Receiver Status Change	RO LH	1 = Local receiver status changed since last read 0 = Interrupt cleared	0
3	Duplex Mode Change	RO LH	1 = Duplex mode changed since last read 0 = Interrupt cleared	0
2	Link Speed Change	RO LH	1 = Link speed changed since last read 0 = Interrupt cleared	0
1	Link Status Change	RO LH	1 = Link status changed since last read 0 = Interrupt cleared	0
0	Receive CRC Error	RO LH	1 = Receive CRC error occurred since last read 0 = Interrupt cleared	0

The INTR LED output is asserted when any bit in 1000BASE-T/100BASE-TX/10BASE-T Interrupt Status register is set and the corresponding bit in the 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask register is cleared.

Signal Detect/Energy Detect Change

This bit indicates the fiber signal detect (bit 4 in register 1Ch, shadow value 11111) or the copper energy detect (bit 5 in register 1Ch, shadow value 11111) has changed since the last read.

Illegal Pair Swap

The NetXtreme II returns a 1 in bit 14 when an uncorrectable pair swap error on the twisted-pair cable has been detected since the last time this register was read; otherwise, it returns a 0.

MDIX Status Change

The NetXtreme II returns a 1 in bit 13 when a link pulse or 100BASE-TX carrier was detected on a different pair than previously detected since the last time this register was read; otherwise, it returns a 0.

Exceeded High Counter Threshold

The NetXtreme II returns a 1 in bit 12 when one or more of the counters in registers 12h–14h is above 32 000; otherwise, it returns a 0.

Exceeded Low Counter Threshold

The NetXtreme II returns a 1 in bit 11 when one or more of the counters in registers 12h–14h is above 128 000; otherwise, it returns a 0.

Auto-Negotiation Page Received

The NetXtreme II returns a 1 in bit 10 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

No HCD Link

When the negotiated HCD is not able to establish a link, bit 9 returns a 1 by the NetXtreme II. The bit is cleared when the register is read.

No HCD

When auto-negotiation returns no HCD, bit 8 returns a 1 by the NetXtreme II. The bit is cleared when the register is read.

Negotiated Unsupported HCD

When the auto-negotiation HCD is not supported by the NetXtreme II, bit 7 returns a 1. The NetXtreme II does not support 100BASE-T4. The bit is cleared when the register is read.

Scrambler Synchronization Error

The NetXtreme II returns a 1 in bit 6 when a scrambler synchronization error has been detected since the last time this register was read; otherwise, it returns a 0.

Remote Receiver Status Change

The NetXtreme II returns a 1 in bit 5 when the remote receiver status has changed since the last time this register was read; otherwise, it returns a 0.

Local Receiver Status Change

The NetXtreme II returns a 1 in bit 4 when the local receiver status has changed since the last time this register was read; otherwise, it returns a 0.

Duplex Mode Change

The NetXtreme II returns a 1 in bit 3 when the duplex mode has changed since the last time this register was read; otherwise, it returns a 0.

Link Speed Change

The NetXtreme II returns a 1 in bit 2 when the link speed has changed since the last time this register was read; otherwise, it returns a 0.

Link Status Change

The NetXtreme II returns a 1 in bit 1 when the link status has changed since the last time this register was read; otherwise, it returns a 0.

Receive CRC Error

The NetXtreme II returns a 1 in bit 0 when a receive CRC error has been detected since the last time this register was read; otherwise, it returns a 0.



1000BASE-T/100BASE-TX/10BASE-T INTERRUPT MASK REGISTER (ADDRESS 1BH)**Table 513: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh)**

Bit	Name	R/W	Description	Default
15	Signal Detect/Energy Detect Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
14	Illegal Pair Swap	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
13	MDIX Status Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
12	Exceeded High Counter Threshold	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
11	Exceeded Low Counter Threshold	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
10	Auto-Negotiation Page Received	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
9	HCD No Link	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
8	No HCD	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
7	Negotiated Unsupported HCD	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
6	Scrambler Synchronization Error	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
5	Remote receiver status Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
4	Local receiver status Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
3	Duplex mode Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
2	Link Speed Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
1	Link Status Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
0	CRC Error	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1

Interrupt Mask Vector

When bit n of the 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask register is written to 1, the interrupt corresponding to the same bit in the Interrupt Status register is masked. The status bits still operate normally when the interrupt is masked, but do not generate an interrupt output. When the bit is written to 0, the interrupt is unmasked.

1000BASE-T/100BASE-TX/10BASE-T REGISTER 1Ch ACCESS

Reading from and writing to 1000BASE-T/100BASE-TX/10BASE-T register 1Ch is through register 1Ch bits[15:10]. The bits[14:10] set the shadow value of register 1Ch, and bit 15 enables the writing of the bits[9:0]. Setting bit 15 allows writing to bits[9:0] of register 1Ch. To read register 1C shadow zzzzz, set writes to register 1Ch with bit 15 = 0, and bits[14:10] to zzzzz first. The subsequent register read from register 1Ch contains the shadow zzzzz register value. All of the register 1Ch shadow values are listed in [Table 582](#).

Table 514: 1000BASE-T/100BASE-TX/10BASE-T Register 1Ch Shadow Values

Shadow Value	Register Name
00010	"1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow Value 00010)" on page 677
00011	"1000BASE-T/100BASE-TX/10BASE-T Clock Alignment Control Register (Address 1Ch, Shadow Value 00011)" on page 679
00100	"1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow 04h)" on page 792
00101	"1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h)" on page 794
01000	"1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow 08h)" on page 796
01001	"1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow 09h)" on page 798
01010	"Auto Power-Down Register (Address 1Ch, Shadow 0Ah)" on page 802
01101	"1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow 0Dh)" on page 806
01110	"1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow 0Eh)" on page 808
01111	"LED GPIO Control/Status Register (Address 1Ch, Shadow Value 0Fh)" on page 810
11000	"1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h)" on page 816
11010	"1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010)" on page 695
11011	"Auxiliary Control Register (Address 1Ch, Shadow 1Bh)" on page 818
11100	"Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow 1Ch)" on page 820
11101	"Misc. 1000BASE-X Status Register (Address 1Ch, Shadow 1Dh)" on page 822
11110	"1000BASE-T/100BASE-TX/10BASE-T Autodetect Medium Register (Address 1Ch, Shadow Value 11110)" on page 704
11111	"Mode Control Register (Address 1Ch, Shadow 1Fh)" on page 824

1000BASE-T/100BASE-TX/10BASE-T SPARE CONTROL 1 REGISTER (ADDRESS 1Ch, SHADOW VALUE 00010)

Table 515: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow Value 00010)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	00010 = Spare Control 1 register	0
13		R/W		0
12		R/W		0
11		R/W		1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Reserved	R/W	Write as 0, ignore when read	0
6	Reserved	R/W	Write as 0, ignore when read	0
5	Reserved	R/W	Write as 0, ignore when read	0
4	100FX Mode	R/W	1 =Enable 100BASE-FX mode on Copper TRD± pins 0 = Normal Copper operation	0
3	Transmit CRC Enable	R/W	1 =Enable Transmit CRC checker 0 = Disable Transmit CRC checker	0
2	Link Speed LED mode	R/W	1 = Enable Link Speed LED mode LINKSPD[2:1] = speed 10: 1000BASE-T link 01: 100BASE-TX link 11: 10BASE-T or Auto-Negotiating 0 = Normal link LED mode	0
1	Reserved	R/W	Write as 0, ignore when read	0
0	Link LED Mode	R/W	1 = Enable link LED mode LINKSPD[2:1] = speed 00: 1000BASE-T link 01: 100BASE-TX link 10: 10BASE-T link or no link SLAVE = Active low 10/100/1000BASE-T link 0 = Normal link LED mode	0

Write Enable

Bit 15 = 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 = 0 and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits 9:0.

Shadow Register Selector

Bits[14:10] of this register must be set to 00010 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 register address 1Ch.

100BASE-FX Mode

Setting bit 4 = 1, it enables 100BASE-FX mode. In addition to bit 4 being set, the PHY also needs to be forced to 100BASE-T full duplex mode (reg. 00h = 2100h) and the rise/fall times of the transmitter set to 0 ns (reg. 18h = 0430h).

Transmit CRC

Setting bit 3 = 1, the transmit CRC checker is enabled. When a transmit CRC error is detected, bit 0 is set = 1 in Expansion Register 01h.

Link Speed LED

Setting bit 2 = 1, the Link Speed Led mode is enabled.

Link LED Mode

Setting bit 0 = 1, enables Link LED mode. The LINKSPD2/LINKSPD1 are Link/Speed LED and SLAVE LED is LINK LED to indicate a link for 10BASE-T, 100BASE-TX or 1000BASE-T. When this bit is cleared, the LINKSPD2, LINKSPD1, and SLAVE operate in their normal mode.

1000BASE-T/100BASE-TX/10BASE-T CLOCK ALIGNMENT CONTROL REGISTER (ADDRESS 1Ch, SHADOW VALUE 00011)

Table 516: 1000BASE-T/100BASE-TX/10BASE-T Clock Alignment Control Register (Address 1Ch, Shadow Value 00011)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	00011 = Clock Alignment Control register	0
13		R/W		0
12		R/W		0
11		R/W		1
10		R/W		1
9	RGMII GTXCLK Clock Delay Enable	R/W	1 = Enable GTXCLK delay 0 = Normal mode (bypass GTXCLK delay)	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Reserved	R/W	Write as 0, ignore when read	0
6	Reserved	R/W	Write as 0, ignore when read	0
5	Reserved	R/W	Write as 0, ignore when read	0
4	Reserved	R/W	Write as 0, ignore when read	0
3	Reserved	R/W	Write as 0, ignore when read	0
2	Reserved	R/W	Write as 0, ignore when read	0
1	Reserved	R/W	Write as 0, ignore when read	0
0	Reserved	R/W	Write as 0, ignore when read	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Clock Alignment register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 00011 to enable read/write to the Clock Alignment Control register address 1Ch.

RGMII GTXCLK Clock Delay Enable

Setting Bit 9 = 1, enables the GTXCLK internal delay of 1.9 ns. When this bit is cleared, the GTXCLK delay is bypassed. This should only be used when in RGMII or RTBI mode.

1000BASE-T/100BASE-TX/10BASE-T SPARE CONTROL 2 REGISTER (ADDRESS 1Ch, SHADOW VALUE 00100)

Table 517: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow Value 00100)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	00100 = Spare Control 2 register	0
13		R/W		0
12		R/W		1
11		R/W		0
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Reserved	R/W	Write as 0, ignore when read	0
6	Reserved	R/W	Write as 0, ignore when read	0
5	Reserved	R/W	Write as 0, ignore when read	0
4	Reserved	R/W	Write as 0, ignore when read	0
3	Reserved	R/W	Write as 1, ignore when read	1
2	Reserved	R/W	Write as 1, ignore when read	1
1	Energy Detect on INTR LED Pin	R/W	1 = Routes Energy Detect to interrupt signal. Use LED selectors (register 1Ch shadow 01101 and 01110) and program to INTR mode. 0 = INTR LED pin performs the Interrupt function.	0
0	Reserved	R/W	Write as 0, ignore when read	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] must be set to 00100 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 register.

Energy Detect on INTR LED Pin

Bit 1 = 1, enables the signal detect or energy detect input on the INTR LED pin. Set the LED selector register to enable INTR LED mode (1Ch shadow 01101 and or 01110 set bits[7:4] and or bits[3:0] to 0110 depending on the LED being used). When Energy is detected (bit 5 = 1 in register 1Ch, shadow value 11111) the INTR pin will be driven high.

1000BASE-T/100BASE-TX/10BASE-T SPARE CONTROL 3 REGISTER (ADDRESS 1Ch, SHADOW VALUE 00101)

**Table 518: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register
(Address 1Ch, Shadow Value 00101)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	00101 = Spare Control 3 register	0
13		R/W		0
12		R/W		1
11		R/W		0
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Reserved	R/W	Write as 0, ignore when read	0
6	TXC Disable	R/W	1 = TXC disabled when in 1000BASE-T mode 0 = TXC enabled when in 1000BASE-T mode	0
5	SD/Energy Detect Change	R/W	1 = Interrupt based on energy detection or signal detect change. 0 = Normal mode	0
4	Reserved	R/W	Write as 1, ignore when read	1
3	Reserved	R/W	Write as 1, ignore when read	1
2	Reserved	R/W	Write as 1, ignore when read	1
1	DLL Auto Power-Down	R/W	1 = Auto power down of DLL is disabled 0 = Auto power down of DLL is enabled	1
0	CLK125 Output	R/W	1 = Enable CLK125 output 0 = Disable CLK125 output	1

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow register values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] must be set to 00101 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 register.

TXC Disable

Setting this bit disables the TXC clock when in 1000BASE-T mode. The TXC clock is still enabled when in 100BASE-TX/FX and 10BASE-T modes.

SD/Energy Detect Change

setting this bit will assert the INTR output pin when there is a change in status in bits[5:4] in register 1Ch, shadow value 11111 and bit 15 =1 in register 1Bh.

DLL Auto Power-Down

Clearing this bit enables the auto power-down mode of the internal DLL. This feature enables additional power savings. This feature should only be used during auto power-down mode.

CLK125 Output

Setting this bit enables the CLK125 output; clearing this bit disables the CLK125 output. When the PHYAREV pin is pulled low at reset, this bit is only valid for the PHY address that is strapped in by the PHYAD[4:0] pins. When the PHYAREV pin is pulled high at reset, this bit is only valid for the PHY address that is strapped in by the PHYAD[4:0] pins + 3.

1000BASE-T/100BASE-TX/10BASE-T LED STATUS REGISTER (ADDRESS 1Ch, SHADOW VALUE 01000)

**Table 519: 1000BASE-T/100BASE-TX/10BASE-T LED Status Register
(Address 1Ch, Shadow Value 01000)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01000 = LED Status register	0
13		R/W		1
12		R/W		0
11		R/W		0
10		R/W		0
9	Reserved	RO	Ignore on read	0
8	SLAVE Indicator	RO	1 = Master mode 0 = Slave mode	0
7	FDX Indicator	RO	1 = Half-duplex mode 0 = Full-duplex mode	0
6	INTR Indicator	RO	1 = No active Interrupt 0 = Interrupt activated	0
5	Reserved	RO	Ignore on read	0
4	LINKSPD Indicator	RO	11 = No link	0
3		RO	10 = 10BASE-T LINK 01 = 100BASE-TX LINK 00 = 1000BASE-T LINK	0
2	Transmit Indicator	RO	1 = No transmit activity 0 = Transmit activity	0
1	Receive Indicator	RO	1 = Not receive activity 0 = Receive activity	0
0	Quality Indicator	RO	1 = Quality not good mode 0 = Quality good mode	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Status register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01000 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T LED Status register.

Slave Indicator

When bit 8 returns a 0, the device is in the slave mode. When this bit returns a 1, the device is not in the slave mode.

FDX Indicator

When bit 7 returns a 0, the device is in the full-duplex mode. When this bit returns a 1, the device is not in the full-duplex mode.

INTR Indicator

When bit 6 returns a 0, the device is in the interrupted mode. When this bit returns a 1, the device is not in the interrupted mode.

LINKSPD Indicator

When bits[4:3] return a 00, the device is in the 1000BASE-TX link mode. When these bits return a 01, the device is in the 100BASE-TX link mode. When these bits return a 10, the device is in the 10BASE-T link mode. When these bits return an 11, the device is not linked.

Transmit Indicator

When bit 2 returns a 0, the device is in the transmitting mode. When this bit returns a 1, the device is not in the transmitting mode.

Receive Indicator

When bit 1 returns a 0, the device is in the receiving mode. When this bit returns a 1, the device is not in the receiving mode.

Quality Indicator

When bit 0 returns a 0, the device is in the quality good mode. When this bit returns a 1, the device is not in the quality good mode.

1000BASE-T/100BASE-TX/10BASE-T LED CONTROL REGISTER (ADDRESS 1Ch, SHADOW VALUE 01001)

**Table 520: 1000BASE-T/100BASE-TX/10BASE-T LED Control Register
(Address 1Ch, Shadow Value 01001)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01001 = LED Control register	0
13		R/W		1
12		R/W		0
11		R/W		0
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Reserved	R/W	Write as 0, ignore when read	0
6	Reserved	R/W	Write as 0, ignore when read	0
5	Override GBIC LED mode	R/W	1 = LEDs not remapped in GBIC mode. 0 = In GBIC mode, LEDs mapped as follows. LED1: RX_LOSS LED2: RX Activity LED3: TX Activity LED4: LINK	0
4	ACTIVITY/LINK LED Enable	R/W	1 = Drive activity/link data on ACTIVITY LED 0 = Drive activity data on ACTIVITY LED	0
3	ACTIVITY LED Enable	R/W	1 = Drive activity data on ACTIVITY LED 0 = Drive receive data on ACTIVITY LED	1
2	Remote Fault LED Enable	R/W	1 = Drive remote fault on quality LED 0 = Normal operation	0
1	Link Utilization LED Selector	R/W	00 = Normal activity (fixed blink rate) 01 = Transmit activity with variable blink rate 10 = Receive activity with variable blink rate 11 = Transmit/receive activity with variable blink rate	0
0			Note: This mode has higher priority than the activity LED enable mode in bit 3	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Control register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01001 to enable read/write to the register address 1Ch.

Override GBIC LED

Bit 5 is automatically set to 0 for GBIC mode and maps RX_LOSS to LED1, RX Activity to LED2, TX Activity to LED3 and LINK to LED4. To override the re-mapping of the LEDs while in GBIC mode, set bit 5 = 1.

ACTIVITY/LINK LED Enable

Setting bit 4 = 1, drives activity/link data on the ACTIVITY LED.

ACTIVITY LED Enable

Setting bit 3 = 1, drives activity data on the ACTIVITY LED. Otherwise, it drives receive data on ACTIVITY LED.

Remote Fault LED Enable

Setting bit 2 =1 drives remote fault on the QUALITY LED.

Link Utilization LED Selector

Bits[1:0] apply to the LED programmed to the ACTIVITY mode only. When in ACTIVITY LED mode, the LED expresses an estimated activity in terms of blink rate. The blink rate of the LED increases as the activity duty cycle increases by increments of 10%. For duty cycles of 0.001 to 10%, the LED blinks at 3 Hz; for duty cycles of 10 to 20%, the LED blinks at 6 Hz; and for duty cycles of 90 to 96%, the LED blinks at 30 Hz. Even though the frequency of the LED blink increases, the duty cycle of the LED stays at about 50%. The ACTIVITY LED can be programmed to display the following:

- 00 = Normal activity (fixed blink rate)
- 01 = Transmit activity with variable blink rate
- 10 = Receive activity with variable blink rate
- 11 = Transmit/receive activity with variable blink rate

1000BASE-T/100BASE-TX/10BASE-T AUTO POWER-DOWN REGISTER (ADDRESS 1Ch, SHADOW VALUE 01010)

Table 521: 1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down Register (Address 1Ch, Shadow Value 01010)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01010 = LED Status register	0
13		R/W		1
12		R/W		0
11		R/W		1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Reserved	R/W	Write as 0, ignore when read	0
6	Reserved	R/W	Write as 0, ignore when read	0
5	Auto Power-Down Mode	R/W	1 = Auto power-down mode enabled 0 = Auto power-down mode disabled	0
4	Sleep Timer Select	R/W	1 = Sleep timer is 5.4 seconds 0 = Sleep timer is 2.7 seconds	0
3	Wake-up Timer Select	R/W	Counter for wake-up timer in units of 84 milliseconds.	0
2		R/W	0001 = 84 milliseconds	0
1		R/W	0010 = 168 milliseconds	0
0		R/W	... 1111 = 1.26 seconds	1

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] must be set to 01010 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down register address 1Ch.

Auto Power-Down Mode.

Setting bit 5 = 1 enables the auto power-down mode.

Sleep Timer Select

Setting bit 4 = 1, changes the wake-up time leaving auto power-down mode.

Wake-Up Timer Select

The port continues wake-up mode for a time based on the count stored in bit[3:0]. The minimum value is 84 milliseconds and the maximum value is 1.26 seconds. This only applies when the part is in auto power-down mode.

**1000BASE-T/100BASE-TX/10BASE-T LED SELECTOR 1 REGISTER (ADDRESS 1Ch,
SHADOW VALUE 01101)**

**Table 522: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register
(Address 1Ch, Shadow Value 01101)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01101 = LED Status register	0
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	LED2 Selector	R/W	0000: <u>LINKSPD[1]</u>	0
6		R/W	0001: <u>LINKSPD[2]</u>	0
5		R/W	0010: <u>XMITLED</u>	0
4		R/W	0011: <u>ACTIVITY</u> 0100: <u>FDXLED</u> 0101: <u>SLAVE</u> 0110: <u>INTR</u> 0111: <u>QUALITY</u> 1000: <u>RCVLED</u> 1001: <u>RESERVED</u> 1010: <u>MULTICOLOR[2]</u> 1011: <u>RESERVED</u> 1100: <u>ENERGYLNK</u> 1101: CRS (SGMII mode) 1110: Off (high) 1111: On (low)	1

**Table 522: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register
(Address 1Ch, Shadow Value 01101) (Cont.)**

Bit	Name	R/W	Description	Default
3	LED1 Selector	R/W	0000: <u>LINKSPD[1]</u> 0001: <u>LINKSPD[2]</u>	0
2		R/W	0010: XMITLED	0
1		R/W	0011: ACTIVITY	0
0		R/W	0100: FDXLED 0101: SLAVE 0110: INTR 0111: QUALITY 1000: RCVLED 1001: RESERVED 1010: MULTICOLOR[1] 1011: RESERVED 1100: ENERGYLNK 1101: CRS (SGMII mode) 1110: Off (high) 1111: On (low)	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Selector register 1 bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01101 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T LED Selector register 1 address 1Ch.

LED2 Selector

Bits[7:4] select the LED2 mode.

LED1 Selector

Bits[3:0] select the LED1 mode.

**1000BASE-T/100BASE-TX/10BASE-T LED SELECTOR 2 REGISTER (ADDRESS 1Ch,
SHADOW VALUE 01110)**

**Table 523: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register
(Address 1Ch, Shadow Value 01110)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01110 = LED Status register	0
13		R/W		1
12		R/W		1
11		R/W		1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	LED4 Selector	R/W	0000: <u>LINKSPD[1]</u> 0001: <u>LINKSPD[2]</u> 0010: <u>XMITLED</u> 0011: <u>ACTIVITY</u> 0100: <u>FDXLED</u> 0101: <u>SLAVE</u> 0110: <u>INTR</u> 0111: <u>QUALITY</u> 1000: <u>RCVLED</u> 1001: <u>RESERVED</u> 1010: <u>MULTICOLOR[2]</u> 1011: <u>RESERVED</u> 1100: <u>ENERGYLNK</u> 1101: CRS (SGMII mode) 1110: Off (high) 1111: On (low)	0
6		R/W		1
5		R/W		1
4		R/W		0

**Table 523: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register
(Address 1Ch, Shadow Value 01110) (Cont.)**

Bit	Name	R/W	Description	Default
3	LED3 Selector	R/W	0000: <u>LINKSPD[1]</u> 0001: <u>LINKSPD[2]</u>	0
2		R/W	0010: XMITLED	0
1		R/W	0011: ACTIVITY	1
0		R/W	0100: FDXLED 0101: SLAVE 0110: INTR 0111: QUALITY 1000: RCVLED 1001: RESERVED 1010: MULTICOLOR[1] 1011: RESERVED 1100: ENERGYLNK 1101: CRS (SGMII mode) 1110: Off (high) 1111: On (low)	1

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits [14:10] of this register must be set to 01110 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 register address 1Ch.

LED4 Selector

Bits[7:4] select the LED4 mode.

LED3 Selector

Bits[3:0] select the LED3 mode.

1000BASE-T/100BASE-TX/10BASE-T LED GPIO CONTROL/STATUS REGISTER (ADDRESS 1Ch, SHADOW VALUE 01111)

Table 524: 1000BASE-T/100BASE-TX/10BASE-T LED GPIO Control/Status Register (Address 1Ch, Shadow Value 01111)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01111 = LED Status register	0
13		R/W		1
12		R/W		1
11		R/W		1
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	LED I/O Status	RO	1 = LED pin is an input	0
6		RO	0 = LED pin is an output	0
5		RO		0
4		RO	Bit 7: LED4 pin status Bit 6: LED3 pin status Bit 5: LED2 pin status Bit 4: LED1 pin status	0
3	Programmable LED I/O Control	R/W	1 = Disable LED output	0
2		R/W	0 = Enable LED output	0
1		R/W		0
0		R/W	Bit 3: LED4 pin control Bit 2: LED3 pin control Bit 1: LED2 pin control Bit 0: LED1 pin control	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED GPIO Control/Status register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits 9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

1000BASE-T/100BASE-TX/10BASE-T LED GPIO Control/Status register bits[14:10] must be set to 01111 to enable read/write to the register address 1Ch.

LED I/O Status

Bits[7:4] read back the status of the LED pin. If the bit = 1, it is an input. If the bit = 0, it is an output.

Programmable LED I/O Control

Setting any of the bits[3:0] = 1 will cause the corresponding LED pin(s) to disable the LED output(s). Clearing any of the bits[3:0] will enable the corresponding LED output(s).

1000BASE-T/100BASE-TX/10BASE-T AUTODETECT SGMII/MEDIA CONVERTER REGISTER (ADDRESS 1Ch, SHADOW VALUE 11000)

Table 525: 1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow Value 11000)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11000 = LED Status register	1
13		R/W		1
12		R/W		0
11		R/W		0
10		R/W		0
9	SerDes Resolution Fault	RO	1 = Selected field mismatch 0 = No mismatch or SGMII/media converter autodetect mode is disabled	0
8	Reserved	RO	Ignore on read	0
7	Reserved	RO	Ignore on read	0
6	Reserved	RO	Ignore on read	0
5	Reserved	RO	Ignore on read	0
4	Reserved	RO	Ignore on read	0
3	Reserved	RO	Ignore on read	0
2	SGMII/GBIC Jumbo Packet Transmission	R/W	1 = Support the transmission of Jumbo packets 0 = Normal mode	1
1	SGMII 10/100 Low RX Latency Mode	R/W	1 = Low RX Latency mode enabled 0 = Normal mode	0
0	SGMII/Media Converter Autodetect Mode Enable	R/W	1 = Enable SGMII/media converter autodetect mode 0 = Normal operation	0

Write Enable

During a write to this register, setting bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] of this register must be set to 11000 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter address 1Ch.

SerDes Resolution Fault

Bit 9 = 1 indicates there is a selected field mismatch on bit 0 of the base page word. Otherwise, it reads a 0.

SGMII/GBIC Jumbo Packet

Bit 2 is automatically set to 1 when in SGMII or GBIC mode. When in this mode Jumbo packets up to 10 KB can be transmitted.

SGMII 10/100 Low RX Latency

Setting bit 1= 1 bypasses the RX fifo when in 10BASE-T or 100BASE-TX mode to reduce latency. This should only be done if the PHY and the Switch's reference clocks are run off the same reference.

SGMII/Media Converter Autodetect Mode Enable

Setting bit 0 = 1 enables an internal PHY algorithm to detect and change the mode of operation (register 1Ch, shadow 11111, bits 2:1) to match the MAC/switch link partner automatically.

1000BASE-X AUTO-NEGOTIATION DEBUG REGISTER (ADDRESS 1Ch, SHADOW VALUE 11010)

Table 526: 1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11010 = LED Status register	1
13		R/W		1
12		R/W		0
11		R/W		1
10		R/W		0
9	Consistency Mismatch	RO	1 = Consistency mismatch occurred since last read	0
		LH	0 = No consistency mismatch occurred since last read	
8	RUDI Invalid	RO	1 = RUDI invalid detected since last read	0
		LH	0 = No RUDI invalid detected since last read	
7	Comma Detected	RO	1 = Comma detected since last read	0
		LH	0 = No comma detected since last read	
6	AN_Sync_Status	RO	1 = AN_Sync_Status has not failed since last read	0
		LL	0 = AN_sync_status failed since last read	
5	Idle Detect State	RO	1 = Idle detect state entered since last read	0
		LH	0 = Idle detect state has not been entered since last read	
4	Complete Acknowledge State	RO	1 = Complete acknowledge state entered since last read	0
		LH	0 = Complete acknowledge state has not been entered since last read	
3	Acknowledge Detect State	RO	1 = Acknowledge detect state entered since last read	0
		LH	0 = Acknowledge detect state has not been entered since last read	
2	Ability Detect State	RO	1 = Ability detect state entered since last read	0
		LH	0 = Ability detect state has not been entered since last read	

Table 526: 1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010) (Cont.)

Bit	Name	R/W	Description	Default
1	Error State When Register 1Ch, shadow value 11011, bit 3 = 1	RO LH	1 = Error state entered since last read 0 = Error state has not been entered since last read	0
	SYNC_Status Failed When Register 1Ch, shadow value 11011, bit 3 = 0	RO LH	1 = Sync_status failed since last read 0 = Sync_status has not been failed since last read	0
0	AN_Enable State	RO LH	1 = AN_Enable state entered since last read 0 = AN_Enable state has not been entered since last read	0

Write Enable

During a write to this register, setting 1000BASE-X Auto-Negotiation Debug register 1 bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow values in bits 14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 11010 to enable read/write to the 1000BASE-X Auto-Negotiation Debug register.

Consistency Mismatch

Bit 9 = 1 indicates a consistency mismatch occurred since last read.

RUDI Invalid

Bit 8 = 1 indicates a RUDI (Rx_UnitData Indicate, specified in IEEE 36.2.5.1.6) invalid detected since last read.

Comma Detected

Bit 7 = 1 indicates a comma was detected since last read.

AN_Sync_Status

Bit 6 = 1 indicates the AN_Sync_Status has not failed since last read.

Idle Detect State

Bit 5 = 1 indicates the idle detect state entered since last read.

Complete Acknowledge State

Bit 4 = 1 indicates the complete acknowledge state entered since last read.

Acknowledge Detect State

Bit 3 = 1 indicates the acknowledge detect state entered since last read.

Ability Detect State

Bit 2 = 1 indicates the ability detect state entered since last read.

Error State

When register 1Ch, shadow value 11011, bit 3 = 1, Bit 1 = 1 in register 1Ch, shadow value 11010 indicates the error state entered since last read.

Sync_Status Failed

When register 1Ch, shadow value 11011, bit 3 = 0, Bit 1 = 1 in register 1Ch, shadow value 11010 indicates the sync_status failed since the last read.

AN_Enable State

Bit 0 = 1 indicates the AN_Enable state entered since last read.

AUXILIARY 1000BASE-X CONTROL REGISTER (ADDRESS 1Ch, SHADOW VALUE 11011)

The following is enabled by register 1Ch with shadow value in bits [14:10] = 11011.

Table 527: Auxiliary 1000BASE-X Control Register (Address 1Ch, Shadow Value 11011)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11011 = LED Status register	1
13		R/W		1
12		R/W		0
11		R/W		1
10		R/W		1
9	Use Fiber (1000BASE-X) Mode Counters	R/W	1 = Use registers 12h–14h for 1000BASE-X data (Fiber) 0 = Use registers 12h–14h for 10BASE-T, 100BASE-TX, 100BASE-FX and 1000BASE-T data (Copper)	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	SGMII/GBIC False Carrier Mode	R/W	1 = Send packets with TXEN, TXER, TXD = 55h for duration of false carrier in SGMII or GBIC half duplex mode 0 = Ignore false carriers in SGMII or GBIC mode	1
6	Disable Carrier Extend	R/W	1 = Force RXER and RXD to zeros in TRR+ extend state (PCS receive state) 0 = Normal operation	0
5	Disable TRRR	R/W	1 = Bypass extend_by_1 state (PCS transmit state) 0 = Normal operation	0

Table 527: Auxiliary 1000BASE-X Control Register (Address 1Ch, Shadow Value 11011) (Cont.)

Bit	Name	R/W	Description	Default
4	Disable Remote Fault Sensing	R/W	1 = Disable automatic remote fault sensing of auto-negotiation resolution error 0 = Normal operation	0
3	Auto-negotiation Error Timer Enable	R/W	1 = Enable auto-negotiation error timer (ability_detect state, acknowledge_detect state, or idle_detect state) 0 = Enable sync_status error	0
2	Comma Detect Enable	R/W	1 = Enable comma detection 0 = Disable comma detection	1
1	FIFO Elasticity (1000BASE-X PCS Transmit) (10/100 SGMII Transmit and Receive)	R/W	1 = High elasticity to support jumbo packets (supports 10/100/1000 jumbo packets) 0 = Low elasticity (low latency)	1
0	Disable Receive CRC Checker	R/W	1 = Disable CRC checker 0 = Enable CRC checker	1

Write Enable

During a write to this register, setting 1000BASE-X Control register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 11011 to enable read/write to the Auxiliary 1000BASE-X Control register.

Use Fiber Mode Counters

Setting bit 9 = 1 enables 1000BASE-X (Fiber) data to be presented on registers 12h–14h. Setting bit 9 = 0 enables 10BASE-T, 100BASE-TX, 100BASE-FX and 1000BASE-TX data to be presented on registers 12h–14h.

SGMII/GBIC False Carrier Mode

Setting bit 7 = 1 causes the data 55h to be sent out for the duration of a false carrier event when in SGMII or GBIC half duplex mode.

Disable Carrier Extend

Setting bit 6 = 1 causes the carrier extend symbol to be replaced with zeros.

Disable TRRR

Setting bit 5 = 1 causes the PHY to transmit only a TRR for odd size packets.

Disable Remote Fault Sensing

Setting bit 4 = 1 disables automatic remote fault sensing of an auto-negotiation resolution error.

Auto-Negotiation Error Timer Enable

When bit 3 = 1 in register 1Ch, SV 11011 and bit 1 = 1 in register 1Ch, SV 11010, it indicates that an auto-negotiation timer error has been detected for the following states:

- Ability_detect state
- Acknowledge_detect state
- Idle_detect state

When bit 3 = 0 in reg. 1Ch, SV 11011 and bit 1 = 1, it indicates that sync_status failure.

Comma Detect Enable

Setting bit 2 = 1 enables comma detection.

FIFO Elasticity

Setting bit 1 = 1 enables jumbo packet reception.

Disable Receive CRC Checker

Setting bit 0 = 1 register disables the Receive CRC checker.



AUXILIARY 1000BASE-X STATUS REGISTER (ADDRESS 1Ch, SHADOW VALUE 11100)**Table 528: Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow Value 11100)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11100 = LED Status register	1
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		0
9	Link Status Change	RO	1 = Link status change has occurred since last read	0
		LH	0 = Link status change has not occurred since last read	
8	SGMII Selector Mismatch	RO	1 = SGMII selector mismatch in SGMII mode 0 = Fiber, Copper, GBIC mode, or SGMII selector does not mismatch, or auto-negotiation is disabled	0
7	Auto-Negotiation Resolution Error	RO	1 = Auto-negotiation HCD is none (no common half-duplex or full-duplex abilities) 0 = SGMII mode, or auto-negotiation disabled, or no resolution error	0
6	Link Partner Remote Fault	RO	Reflects 1000BASE-X register 05h [13:12]	0
5		RO	00 = No remote fault 10 = Off line 01 = Link fault 11 = Auto-negotiation error	0
4	Auto-Negotiation Page Received	RO	1 = Page has been received since last read	0
		LH	0 = Page has not been received since last read	
3	Current Operating Duplex Mode	RO	1 = Phy is operating in full-duplex mode 0 = Phy is operating in half-duplex mode (or auto-negotiation has not completed)	0
2	Link Status	RO	1 = Link is up on Fiber side 0 = Link is down on Fiber side	0
1	PAUSE Resolution—Receive Side	RO	1 = Enable pause receive 0 = Disable pause receive	0
0	PAUSE Resolution—Transmit Side	RO	1 = Enable pause transmit 0 = Disable pause transmit	0

Write Enable

During a write to this register, setting Auxiliary 1000BASE-X Status register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 11100 to enable read/write to the Auxiliary 1000BASE-X Status register

Link Status Change

Bit 9 = 1 indicates that the link status has changed since the last register read.

SGMII Selector Mismatch

Bit 8 = 1 indicates an SGMII selector mismatch in SGMII mode.

Auto-Negotiation Resolution Error

Bit 7 = 1 indicates auto-negotiation HCD is none (no common half-duplex or full-duplex abilities).

Link Partner Remote Fault

Bits[6:5] indicates the link partner's remote fault status reflected from 1000BASE-X register 05h bits[13:12].

Auto-Negotiation Page Received

Bit 4 = 1 indicates auto-negotiation page has been received since last read.

Current Operating Duplex Mode

Bit 3 = 1 indicates the PHY is operating in full-duplex mode.

Link Status

Bit 2 = 1 indicates the PHY link is up on the SerDes side.

PAUSE Resolution—Receive Side

Bit 1 = 1 indicates receive pause resolution.

PAUSE Resolution—Transmit Side

Bit 0 = 1 indicates transmit pause resolution.

MISC. 1000BASE-X STATUS REGISTER (ADDRESS 1Ch, SHADOW VALUE 11101)**Table 529: Misc. 1000BASE-X Status Register (Address 1Ch, Shadow Value 11101)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11101 = LED Status register	1
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		1
9	TX FIFO Error	RO LH	1 = Transmit FIFO error since last read 0 = No transmit FIFO error since last read	0
8	RX FIFO Error	RO LH	1 = Receive FIFO error since last read 0 = No receive FIFO error since last read	0
7	Reserved	RO LH	Ignore on read	0
6	Reserved	RO LH	Ignore on read	0
5	False Carrier detected on the SGIN+/- input	RO LH	1 = False carrier detected since last read 0 = No false carriers detected since last read	0
4	Receive CRC Error detected on the SGIN+/- input	RO LH	1 = Receive CRC Error detected since last read 0 = No Receive CRC error detected since last read or mode is disabled via register 1Ch, shadow 11011, bit 0	0
3	Transmit Error detected on the SGOUT+/- output	RO LH	1 = Transmit error code detected since last read (rx_data_error state in PCS receive) 0 = No transmit error code detected since last read	0
2	Receive Error detected on the SGIN+/- input	RO LH	1 = Receive error since last read (early_end state in PCS receive) 0 = No receive error since last read	0
1	Carrier Extend Error Detected	RO LH	1 = Carrier extend error since last read (extend_err state in PCS receive) 0 = No carrier extend error since last read	0
0	Early End Extension Detected	RO LH	1 = Early end extension since last read (early_end_ext state in PCS receive) 0 = No early end extension since last read	0

Write Enable

During a write to this register, setting Misc. 1000BASE-X Status register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] must be set to 11101 to enable read/write to the Misc. 1000BASE-X Status register.



TX FIFO Error

Bit 9 = 1 indicates a transmit FIFO error since the last read.

RX FIFO Error

Bit 8 = 1 indicates a receive FIFO error since the last read.

False Carrier Detected

Bit 5 = 1 indicates a false carrier detected since the last read. False Carrier Detect error is with respect to the PHY's SerDes Receiver (SGIN+/- pins).

CRC Error Detected

Bit 4 = 1 indicates a CRC error detected since the last read. CRC error detection is with respect to the PHY's SerDes Receiver (SGIN+/- pins).

Transmit Error Detected

Bit 3 = 1 indicates a transmit error code detected since the last read. The transmit error detection is with respect to the PHY's SerDes Transmitter (SGOUT+/- pins).

Receive Error Detected

Bit 2 = 1 indicates a receive error code detected since the last read. Receive error detection is with respect to the PHY's SerDes Receiver (SGIN+/- pins).

Carrier Extend Error Detected

Bit 1 = 1 indicates a carrier extend error since the last read. Carrier Extend detection error is with respect to the PHY's SerDes Receiver (SGIN+/- pins).

Early End Extension Detected

Bit 0 = 1 indicates an early end extension since the last read.

1000BASE-T/100BASE-TX/10BASE-T AUTODETECT MEDIUM REGISTER (ADDRESS 1Ch, SHADOW VALUE 11110)

Table 530: 1000BASE-T/100BASE-TX/10BASE-T Autodetect Medium Register (Address 1Ch, Shadow Value 11110)

Register /Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11110 = LED Status register	1
13		R/W		1
12		R/W		1
11		R/W		1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read	0
8	SD Invert	R/W	1 = Fiber Signal Detect is active low from optical module (RX_LOS) 0 = Fiber Signal Detect is active high from optical module (SD) Note: This is only valid for 1000BASE-X mode. This does not work in 100BASE-FX mode.	0
7	Fiber In-Use LED Mode	R/W	1 = Drive transmit LED active low when Fiber is selected, inactive when copper is selected 0 = Normal transmit LED operation	0
6	Fiber LED Mode	R/W	1 = Use Fiber transmit, receive and link for LED's whenever Fiber mode is selected via register 1Ch, shadow value 11111, bits[2:1] = 01 0 = Always use copper transmit, receive and link for LEDs regardless of the mode selected	0
5	Qualify Fiber Signal Detect with Sync Status	R/W	1 = Sync status must be set in order for the fiber signal detect to be active. 0 = Fiber signal detect from pin is used directly.	1
4	Reserved	R/W	Write as 0, ignore when read	0
3	Reserved	R/W	Write as 0, ignore when read	0
2	Auto-Detect Media Default	R/W	1 = Fiber selected when no medium is active 0 = Copper selected when no medium is active	0
1	Auto-Detect Medium Priority	R/W	1 = Fiber selected when both media are active 0 = Copper selected when both media are active	1
0	Auto-Detect Medium Enable	R/W	1 = Enable auto-detect medium 0 = Disable auto-detect medium	0

Write Enable

During a write to this register, setting the 1000BASE-T/100BASE-TX/10BASE-T Autodetect Medium register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] of this register must be set to 11110 to enable the drive transmit LED active low when SerDes is selected; inactive when copper is selected.

SD Invert

Setting bit 8 = 1 allows the PHY to receive RX_LOS signals from optical modules. Setting bit 8 = 1 allows the PHY to receive SD signals from optical modules. This bit is only valid for 1000BASE-X operation. This does not invert the signal when in 100BASE-FX mode.

Fiber In-Use LED Mode

Setting bit 7 = 1 drives the TRANSMIT LED active low when Fiber is selected; inactive when Copper is selected.

Fiber LED Mode

Setting bit 6 = 1 enables the Fiber activity, transmit, receive, and link to be outputted on the ACTIVITY, TRANSMIT, RECEIVE, LINK1, and LINK2 LEDs respectively whenever Fiber mode is selected via register 1Ch, shadow 11111, bits[2:1].

Qualify Fiber Signal Detect with Sync Status

Setting bit 5 = 1, ANDs the SD or RX_LOS input on LED4 with PHY's internal PCS Sync Status. If valid signal SD or RX_LOS signal is received and sync status is true, then bit 4 in register 1Ch, shadow value 11111 will be set = 1. This ensures that the Fiber Signal Detect bit in register 1Ch, shadow value 11111 is only set when a the PHY is receiving valid data and a valid signal on LED4 is active.

Autodetect Media Default

Setting bit 2 = 1 enables the selection of Fiber as the default medium when no medium is active. Clearing this bit sets Copper as the default medium.

Autodetect Media Priority

Setting bit 1 = 1 enables the selection of Fiber as priority when both media are active. Clearing this bit sets Copper as the default medium selection.

Autodetect Media Enable

Setting bit 0 =1 enables the autodetect media function.

**1000BASE-T/100BASE-TX/10BASE-T MODE CONTROL REGISTER (ADDRESS 1Ch,
SHADOW VALUE 11111)**

**Table 531: 1000BASE-T/100BASE-TX/10BASE-T Mode Control Register
(Address 1Ch, Shadow Value 11111)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11111 = LED Status register	1
13		R/W		1
12		R/W		1
11		R/W		1
10		R/W		1
9	Reserved	RO	Ignore on read	0
8	Mode Select Change	RO LH	1 = Interface Mode Select status changed since last read 0 = Interface Mode Select status did not change since last read	0
7	Copper Link	RO	1 = Link is good on the copper interface 0 = Copper link is down	0
6	Fiber Link	RO	1 = Link is good on the SerDes interface when in fiber, SGMII, or GBIC mode 0 = Fiber link is down	0
5	Copper Energy Detect	RO	1 = Energy detected on the copper interface 0 = Energy not detected on the copper interface	0
4	Fiber Signal Detect	RO	1 = Filtered energy detected on the SerDes interface 0 = Energy not detected on the SerDes interface	0
3	Reserved	RO	Ignore on read	1
2	Mode Select	R/W	00 = Copper	00
1		R/W	01 = Fiber 10 = SGMII 11 = GBIC (Media Converter)	
0	Enable 1000BASE-X Registers	R/W	1 = Select 1000BASE-X registers for addresses 00h–0Fh 0 = Select copper registers for addresses 00h–0Fh	00

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Mode Control register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] of this register must be set to 11111 to enable read/write to the Mode Control register address 1Ch.

Mode Select Change

Bit 8 = 1 indicates that there is change in the interface mode selection. Otherwise, it reads a 0.

Copper Link

Bit 7 = 1 indicates that the link status of the copper interface is up. Otherwise, it reads a 0.

Fiber Link

When in:

- Fiber or GBIC mode, Bit 6 = 1 indicates that the Fiber link is up.
- When in SGMII mode, Bit 6 = 1 indicates that the SGMII interface is linked.

Copper Energy Detect

Bit 5 = 1 indicates that energy (link pulses, FLPs or data) is detected on the copper interface.

Fiber Signal Detect

Bit 4 = 1 indicates that either a SD or RX_LOS signal from an optical module is detected on the LED4 pin. If bit 5 = 1 in register 1Ch, shadow value 11110, then the input to the LED4 pin is ANDed with sync status before bit 4 gets set = 1.

Mode Select

Bits[2:1] selects one of the four available interfaces (Copper, Fiber, SGMII, GBIC).

Enable 1000BASE-X Registers

Setting bit 0 = 1 enables the 1000BASE-X register set for addresses 00h–0Fh. Clearing Bit 0 of the Mode Control register enables the copper register set for addresses 00h–0Fh.

1000BASE-T/100BASE-TX/10BASE-T MASTER/SLAVE SEED REGISTER (ADDRESS 1DH) BIT 15 = 0

Table 532: 1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0

Bit	Name	R/W	Description	Default
15	Enable Shadow Register	R/W	1 = Select shadow register 0 = Normal operation Writes to the selected register are done on a single cycle.	0
14	Master/Slave Seed Match	RO LH	1 = Seeds match 0 = Seeds do not match	0
13	Link Partner Repeater/DTE Bit	RO	1 = Link partner is a repeater/switch device 0 = Link partner is a DTE device	0
12	Link Partner Manual Master/ Slave Configuration Value	RO	1 = Link partner is configured as master 0 = Link partner is configured as slave	0
11	Link Partner Manual Master/ Slave Configuration Enable	RO	1 = Link partner manual master/slave configuration enabled 0 = Link partner manual master/slave configuration disabled	0
10	Local Master/Slave Seed Value	R/W	Returns the automatically generated master/slave random seed.	0
9		R/W		0
8		R/W		0
7		R/W		0
6		R/W		0
5		R/W		0
4		R/W		0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		0

Enable Shadow Register

When bit 15 = 0, the Master/Slave Seed register is selected. If bit 15 = 1, the shadow register HCD Status register is selected for read/write.

Master/Slave Seed Match

When bit 14 returns a 1 when the master/slave seed matches; otherwise, it returns a 0.

Link Partner Repeater/DTE Bit

When bit 13 returns a 1, it indicates that the link partner is configured as a repeater or a switch. If this bit returns a 0, it indicates that the link partner is configured as a DTE port.

Link Partner Manual Master/Slave Configuration Value

When bit 12 returns a 1, it indicates that the link partner is configured as a master. If this bit returns a 0, it indicates that the link partner is configured as a slave.

Link Partner Manual Master/Slave Configuration Enable

When bit 11 returns a 1, it indicates that the link partner's manual master/slave configuration is enabled. If this bit returns a 0, the link partner manual master/slave configuration is disabled.

Local Master/Slave Seed Value

Bits[10:0] return the automatically-generated local master/slave seed value.

1000BASE-T/100BASE-TX/10BASE-T HCD STATUS REGISTER (ADDRESS 1DH) BIT 15 = 1

Table 533: 1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1

Bit	Name	R/W	Description	Default
15	Enable Shadow Register	R/W	1 = Select Shadow register 0 = Normal operation	0
14	Reserved	RO	Ignore on read	0
13	Reserved	RO	Ignore on read	0
12	Reserved	RO LH	Ignore on read	0
11	HCD 1000BASE-T Full Duplex	RO LH	1 = Gigabit full-duplex occurred since last read 0 = HCD cleared	0
10	HCD 1000BASE-T Half Duplex	RO LH	1 = Gigabit half-duplex occurred since last read 0 = HCD cleared	0
9	HCD 100BASE-TX Full Duplex	RO LH	1 = 100BASE-TX full-duplex occurred since last read 0 = HCD cleared	0
8	HCD 100BASE-T Half Duplex	RO LH	1 = 100BASE-TX half-duplex occurred since last read 0 = HCD cleared	0
7	HCD 10BASE-T Full Duplex	RO LH	1 = 10BASE-T full-duplex occurred since last read 0 = HCD Cleared	0
6	HCD 10BASE-T Half Duplex	RO LH	1 = 10BASE-T half-duplex occurred since last read 0 = HCD cleared	0
5	HCD 1000BASE-T Full Duplex (Link Never Came Up)	RO LH	1 = Gigabit full-duplex HCD and <i>link never came up</i> occurred since the last read 0 = HCD cleared	0
4	HCD 1000BASE-T Half Duplex (Link Never Came Up)	RO LH	1 = Gigabit half-duplex HCD and <i>link never came up</i> occurred since the last read 0 = HCD cleared	0

Table 533: 1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1 (Cont.)

Bit	Name	R/W	Description	Default
3	HCD 100BASE-TX Full Duplex (Link Never Came Up)	RO LH	1 = 100BASE-TX full-duplex HCD and <i>link never came up</i> occurred since the last read 0 = HCD cleared	0
2	HCD 100BASE-T Half Duplex (Link Never Came Up)	RO LH	1 = 100BASE-TX half-duplex HCD and <i>link never came up</i> occurred since the last read 0 = HCD cleared	0
1	HCD 10BASE-T Full Duplex (Link Never Came Up)	RO LH	1 = 10BASE-T full-duplex HCD and <i>link never came up</i> occurred since the last read 0 = HCD cleared	0
0	HCD 10BASE-T Half Duplex (Link Never Came Up)	RO LH	1 = 10BASE-T half-duplex HCD and <i>link never came up</i> occurred since the last read 0 = HCD cleared	0



Note: Bits[12:0] are also cleared when auto-negotiation is disabled via MII register 00h, bit 12 = 1, or restarted via MII register 00h, bit 9 = 1.

Enable Shadow Register

When bit 15 = 0, the 1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed register is selected. If bit 15 = 1, the shadow register HCD Status register (auto-negotiation highest common denominator resolution), is selected for read/write. Bit 15 = 1 to be able to read/write to the HCD Status register.

HCD 1000BASE-T FDX

When bit 11 returns a 1, it indicates that a Gigabit full-duplex HCD has occurred since the last read.

HCD 1000BASE-T

When bit 10 returns a 1, it indicates that a Gigabit half-duplex HCD has occurred since the last read.

HCD 100BASE-TX FDX

When bit 9 returns a 1, it indicates that a 100BASE-TX full-duplex HCD has occurred since the last read.

HCD 100BASE-T

When bit 8 returns a 1, it indicates that a 100BASE-TX half-duplex HCD has occurred since the last read.

HCD 10BASE-T FDX

When bit 7 returns a 1, it indicates that a 10BASE-T full-duplex HCD has occurred since the last read.

HCD 10BASE-T

When bit 6 returns a 1, it indicates that a 10BASE-T half-duplex HCD has occurred since the last read.

HCD 1000BASE-T FDX (Link Never Came Up)

When bit 5 returns a 1, it indicates that a Gigabit full-duplex HCD has occurred, but the link was not established since the last read.

HCD 1000BASE-T (Link Never Came Up)

When bit 4 returns a 1, it indicates that a Gigabit half-duplex HCD has occurred, but the link was not established since the last read.

HCD 100BASE-TX FDX (Link Never Came Up)

When bit 3 returns a 1, it indicates that a 100BASE-TX full-duplex HCD has occurred, but the link was not established since the last read.

HCD 100BASE-TX (Link Never Came Up)

When bit 2 a 1, it indicates that a 100BASE-TX half-duplex HCD has occurred, but the link was not established since the last read.

HCD 10BASE-T FDX (Link Never Came Up)

When bit 1 returns a 1, it indicates that a 10BASE-T full-duplex HCD has occurred, but the link was not established since the last read.

HCD 10BASE-T (Link Never Came Up)

When bit 0 returns a 1, it indicates that a 10BASE-T half-duplex HCD has occurred, but the link was not established since the last read.

1000BASE-T/100BASE-TX/10BASE-T TEST 1 REGISTER (ADDRESS 1EH)**Table 534: 1000BASE-T/100BASE-TX/10BASE-T Test 1 Register (Address 1Eh)**

Bit	Name	R/W	Description	Default
15	CRC Error Counter Selector	R/W	1 = Receiver NOT_OK Counters (register 14h) becomes 16 bit CRC error counter (CRC errors are counted only after this bit is set) 0 = Normal operation	0
14	Transmit Error Code Visibility	R/W	1 = False Carrier Sense Counters (register 13h) counts packets received with transmit error codes. 0 = Normal operation	0
13	Reserved	R/W	Write as 0, ignore when read	0
12	Force Link	R/W	1 = Force Link State machine into link pass state 0 = Normal Operation	0
11	Reserved	R/W	Write as 0, ignore when read	0
10	Reserved	R/W	Write as 0, ignore when read	0
9	Reserved	R/W	Write as 0, ignore when read	0
8	Reserved	R/W	Write as 0, ignore when read	0
7	Manual Swap MDI State	R/W	1 = Manually swap MDI state 0 = Normal operation	0
6	Reserved	R/W	Write as 0, ignore when read	0
5	Reserved	R/W	Write as 0, ignore when read	0
4	Reserved	R/W	Write as 0, ignore when read	0
3	Reserved	R/W	Write as 0, ignore when read	0
2	Reserved	R/W	Write as 0, ignore when read	0
1	Reserved	R/W	Write as 0, ignore when read	0
0	Reserved	R/W	Write as 0, ignore when read	0

CRC Error Counter Selector

Setting bit 15 = 1, enables the counting register 14h to count and store the number of CRC errors.

Packets Received with Transmit Error Codes Counter Selector

Setting bit 14 = 1, enables register 13h to start counting packets with transmit error codes and store the counts in register 13h.

Force Link

Setting bit 12 = 1, forces the link state machine into the link pass state.

Manual Swap MDI State

Setting bit 7 = 1, manually swaps the MDI transmit and receive pairs during forced 100BASE-TX and 10BASE-T operation. When this bit is set, the NetXtreme II transceiver transmits on pairs TRD± {1} and receives on TRD± {0} when operating in 100BASE-TX and 10BASE-T modes. If this bit is cleared, the NetXtreme II transmits on pairs TRD± {0} and receives on TRD± {1} when operating in 100BASE-TX and 10BASE-T modes. This bit is ignored when auto-negotiation is enabled. To change the MDI state when in forced 100BASE-TX mode, the PHY must first be put into a non-link condition, then set bit 7 = 1, and finally set the PHY into forced 100BASE-TX mode.



1000BASE-X REGISTERS DESCRIPTIONS

1000BASE-X MII CONTROL REGISTER (ADDRESS 00H)

The following 1000BASE-X registers are enabled by writing to “[Mode Control Register \(Address 1Ch, Shadow 1Fh\)](#)” bit 0 = 1, or when the device is powered up in SerDes mode.

Table 535: 1000BASE-X MII Control Register (Address 00h)

Bit	Name	R/W	Description	Default
15	Reset	R/W SC	1 = PHY reset 0 = Normal operation	0
14	Internal Loopback	R/W	1 = Loopback mode 0 = Normal operation	0
13	Reserved	RO	Write as 0, ignore on read	0
12	Auto-Negotiation Enable	R/W	1 = Auto-negotiation enabled 0 = Auto-negotiation disabled	1
11	Power Down	R/W	1 = Power down 0 = Normal operation	0
10	Isolate	R/W	1 = Electrically isolate PHY from GMII/ RGMII/RTBI 0 = Normal operation	0
9	Restart Auto-Negotiation	R/W SC	1 = Restarting auto-negotiation 0 = Auto-negotiation restart complete	0
8	Duplex Mode	R/W	1 = Full-duplex 0 = Half-duplex	1
7	Collision Test Enable	R/W	1 = Enable the collision test mode 0 = Disable the collision test mode	0
6	Reserved	RO	Write as 1, ignore on read	1
5	Reserved	RO	Write as 0, ignore on read	0
4	Reserved	RO	Write as 0, ignore on read	0
3	Reserved	RO	Write as 0, ignore on read	0
2	Reserved	RO	Write as 0, ignore on read	0
1	Reserved	RO	Write as 0, ignore on read	0
0	Reserved	RO	Write as 0, ignore on read	0

Reset

To reset the NetXtreme II transceiver by software control, a 1 must be written to bit 15 of the 1000BASE-X MII Control register. This bit clears itself after the reset process is complete, and does not need to be cleared using a second MII write. Writes to other 1000BASE-X MII Control register bits have no effect until the reset process is completed, which requires approximately 2.0 microseconds. Writing a 0 to this bit has no effect. A 1 is returned when this bit is read during the reset process; otherwise, it returns a 0.

Internal Loopback

The NetXtreme II can be placed into internal loopback mode by setting bit 14 = 1. The loopback mode can be cleared by writing a 0 to bit 14, or by resetting the PHY.

Auto-Negotiation Enable

When bit 12 = 1, the NetXtreme II mode of operation is controlled by auto-negotiation. When this bit is cleared, the NetXtreme II mode of operation is determined by the duplex mode.

Power Down

When bit 11 = 1, the NetXtreme II is placed into low power standby mode. This bit is set to a 1 when the PHY is in Copper mode.

Isolate

The NetXtreme II can be isolated from the GMII/RGMII/RTBI bus by setting bit 10 = 1. All GMII/RGMII/RTBI outputs are tri-stated, and all RGMII inputs are ignored. Because the management interface is still active, the isolate mode can be cleared by writing a 0 to bit 10 or resetting the PHY.

Restart Auto-Negotiation

Setting bit 9 = 1 forces the auto-negotiation process to be restarted, regardless of the current state of the auto-negotiation state machine.

Duplex Mode

When auto-negotiation is disabled, the duplex mode of the NetXtreme II can be controlled by writing to bit 8. Setting bit 8 = 1 forces the NetXtreme II into full-duplex operation, and clearing this bit forces the NetXtreme II into half-duplex operation. When this bit is read, it returns the last value written. The default value of this bit is determined by the FDX pin at reset.

Collision Test

The NetXtreme II can be placed into collision test mode by setting to bit 7 = 1. In this mode, the COL pin is asserted whenever the TX_EN pin is driven high. The collision test mode can be cleared by writing a 0 to bit 7, or resetting the chip.

1000BASE-X MII STAUTS REGISTER (ADDRESS 01H)

The following 1000BASE-X registers are enabled by writing to "[Mode Control Register \(Address 1Ch, Shadow 1Fh\)](#)" bit 0 = 1.

Table 536: 1000BASE-X MII Status Register (Address 01h)

Bit	Name	R/W	Description	Default
15	100BASE-T4 Capable	RO L	1 = 100BASE-T4 capable 0 = Not 100BASE-T4 capable	0
14	100BASE-X Full-Duplex Capable	RO L	1 = 100BASE-X full-duplex capable 0 = Not 100BASE-X full-duplex capable	0
13	100BASE-X Half-Duplex Capable	RO L	1 = 100BASE-X half-duplex capable 0 = Not 100BASE-X half-duplex capable	0
12	10BASE-T Full-Duplex Capable	RO L	1 = 10BASE-T full-duplex capable 0 = Not 10BASE-T full-duplex capable	0
11	10BASE-T Half-Duplex Capable	RO L	1 = 10BASE-T half-duplex capable 0 = Not 10BASE-T half-duplex capable	0
10	100BASE-T2 Full-Duplex Capable	RO L	1 = 100BASE-T2 full-duplex capable 0 = Not 100BASE-T2 full-duplex capable	0
9	100BASE-T2 Half-Duplex Capable	RO L	1 = 100BASE-T2 half-duplex capable 0 = Not 100BASE-T2 half-duplex capable	0
8	Extended Status	RO H	1 = Extended status information in reg 0Fh 0 = No extended status information in reg 0Fh	1
7	Reserved	RO	Ignore on read	0
6	Management Frames Preamble Suppression	RO H	1 = Preamble can be suppressed 0 = Preamble always required	1
5	Auto-Negotiation Complete	RO	1 = Auto-negotiation complete 0 = Auto-negotiation in progress	0
4	Remote Fault	RO LH	1 = Remote fault detected 0 = No remote fault detected	0
3	Auto-Negotiation Ability	RO H	1 = Auto-negotiation capable 0 = Not auto-negotiation capable	1
2	Link Status	RO LL	1 = Link is up (link pass state) 0 = Link is down (link fail state)	0
1	Jabber Detect	RO L	1 = Jabber condition detected 0 = No jabber condition detected	0
0	Extended Capability	RO H	1 = Extended register capabilities 0 = No extended register capabilities	1

100BASE-T4 Capable

The NetXtreme II is not capable of 100BASE-T4 operation, and returns a 0 when bit 15 is read.

100BASE-X Full-Duplex Capable

The NetXtreme II is not capable of 100BASE-X full-duplex operation, and returns a 0 when bit 14 is read.



100BASE-X Half-Duplex Capable

The NetXtreme II is not capable of 100BASE-X half-duplex operation, and returns a 0 when bit 13 is read.

10BASE-T Full-Duplex Capable

The NetXtreme II is not capable of 10BASE-T full-duplex operation, and returns a 0 when bit 12 is read.

10BASE-T Half-Duplex Capable

The NetXtreme II is not capable of 10BASE-T half-duplex operation, and returns a 0 when bit 11 is read.

100BASE-T2 Full-Duplex Capable

The NetXtreme II is not capable of 100BASE-T2 full-duplex operation, and returns a 0 when bit 10 is read.

100BASE-T2 Half-Duplex Capable

The NetXtreme II is not capable of 100BASE-T2 half-duplex operation, and returns a 0 when bit 9 is read.

Extended Status

The NetXtreme II contains an IEEE Extended Status register at address 0Fh, and returns a 1 when bit 8 is read.

Management Frames Preamble Suppression

The NetXtreme II accepts MII management frames whether or not they are preceded by the preamble pattern, and returns a 1 when bit 6 is read.



Note: Preamble is still required on the first read or write. Preamble suppression can not be disabled.

Auto-Negotiation Complete

The NetXtreme II returns a 1 in bit 5 when auto-negotiation has completed, and the contents of registers 04h, 05h, and 06h are valid. This bit returns a 0 while auto-negotiation is in progress.

Remote Fault

The NetXtreme II returns a 1 in bit 4 when its link partner has signaled a remote fault condition. When a remote fault occurs, the bit is set = 1 and remains so until the remote fault condition has been cleared and the register is read.

Auto-Negotiation Ability

Even if the auto-negotiation function has been disabled, the NetXtreme II is capable of performing IEEE Auto-Negotiation and returns a 1 when bit 3 is read.

Link Status

The NetXtreme II returns a 1 in bit 2 when the link monitor is in the link pass state (indicating that a valid link has been established), otherwise it returns a 0. When a link failure occurs, the Link Status bit is latched at 0 and remains so until the bit is read and the NetXtreme II is in the link pass state.

Jabber Detect

The jabber detect function is not supported for the 1000BASE-X. This bit always returns a 0.

Extended Capability

The NetXtreme II supports extended capability registers, and returns a 1 when bit 0 of the 1000BASE-X MII Status register is read.

1000BASE-X AUTO-NEGOTIATION ADVERTISEMENT REGISTER (ADDRESS 04H)

The following 1000BASE-X registers are enabled by writing to “[Mode Control Register \(Address 1Ch, Shadow 1Fh\)](#)” bits[2:0] = 011.

Table 537: 1000BASE-X Auto-Negotiation Advertisement Register (Address 04h)

Bit	Name	R/W	Description	Default
15	Reserved	RO	Write as 0, ignore on read	0
14	Reserved	RO	Write as 0, ignore on read	0
13	Remote Fault	R/W	00 = No remote fault	0
12		R/W	01 = Link fault 10 = Off line 11 = Auto-negotiation error	0
11	Reserved	R/W	Write as 0, ignore on read	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Reserved	R/W	Write as 0, ignore on read	0
8	Pause	R/W	00 = No pause	0
7		R/W	01 = Symmetric pause 10 = Asymmetrical pause toward link partner 11 = Both asymmetric and symmetric pause toward local device	0
6	Half-Duplex Capable	R/W	1 = 1000BASE-X half-duplex advertised 0 = 1000BASE-X half-duplex not advertised	1
5	Full-Duplex Capable	R/W	1 = 1000BASE-X full-duplex advertised 0 = 1000BASE-X full-duplex not advertised	1
4	Reserved	R/W	Write as 0, ignore on read	0
3	Reserved	R/W	Write as 0, ignore on read	0
2	Reserved	R/W	Write as 0, ignore on read	0
1	Reserved	R/W	Write as 0, ignore on read	0
0	Reserved	R/W	Write as 0, ignore on read	0

Remote Fault

Setting bits[13:12] sends a remote fault indication to the link partner during auto-negotiation. Writing a 00 to this bit clears the Remote Fault transmission bit. This bit returns a nonzero when advertising remote fault; otherwise, it returns a 00.

Pause

Bits[8:7] determine the pause capability of the Switch/MAC that the NetXtreme II is connected to.

Half-Duplex Capable

When bit 6 = 1, the NetXtreme II advertises 1000BASE-X half-duplex capability. When the bit is cleared, the NetXtreme II advertises no 1000BASE-X half-duplex capability. This bit will default to 0 when in GBIC mode and Auto-Negotiation is turned off (INTF_SEL[3:2] = 11 and ANEN = 0); otherwise, this bit will default to 1.

Full-Duplex Capable

When bit 5 = 1, the NetXtreme II advertises 1000BASE-X full-duplex capability. When this bit is cleared, the NetXtreme II advertises no 1000BASE-X full-duplex capability.

SGMII AUTO-NEGOTIATION ADVERTISEMENT REGISTER (ADDRESS 04H)

The following 1000BASE-X registers are enabled by writing to “[Mode Control Register \(Address 1Ch, Shadow 1Fh\)](#)” bits[2:0] = 101.

Table 538: SGMII Auto-Negotiation Advertisement Register (Address 04h)

Bit	Name	R/W	Description	Default
15	Copper Link	R/W	1 = Copper Link 0 = No Link	0
14	Acknowledge	R/W	Reserve for Auto-Negotiation acknowledge as specified in 802.3z	0
13	Reserved	R/W	Ignore on read (Switch/MAC sends 0 only)	0
12	Copper Duplex	R/W	1 = Full Duplex 0 = Half Duplex	0
11	Copper Speed	R/W	11 = Reserved	0
10		R/W	10 = 1000BASE-TX 01 = 100BASE-TX 00 = 10BASE-T	0
9	Reserved	R/W	Ignore on read	0
8	Reserved	R/W	Ignore on read	0
7	Reserved	R/W	Ignore on read	0
6	Reserved	R/W	Ignore on read	0
5	Reserved	R/W	Ignore on read	0
4	Reserved	R/W	Ignore on read	0
3	Reserved	R/W	Ignore on read	0
2	Reserved	R/W	Ignore on read	0
1	Reserved	R/W	Ignore on read	0
0	SGMII Selector	R/W	1 = SGMII mode	1

Note: When SGMII mode is enabled, reading registers 04h reflects the negotiated link, speed, and duplex ability of the NetXtreme II and its link partner's Copper interface. The values written to the register are stored, but not used.

Copper Link

When bit 15 = 1, the NetXtreme II has established a 10BASE-T, 100BASE-TX or 1000BASE-T link.

Copper Duplex

When bit 12 = 1, the NetXtreme II is in Full Duplex mode.

Copper Duplex

When bits[11:10] = 00, the NetXtreme II is in 10BASE-T mode. When bits[11:10] = 01, the NetXtreme II is in 100BASE-TX mode. When bits[11:10] = 10, the NetXtreme II is in 1000BASE-T mode.

SGMII Selector

Bit 0 always returns a 1 when the part is in SGMII mode.

1000BASE-X AUTO-NEGOTIATION LINK PARTNER ABILITY REGISTER (ADDRESS 05H)

The following 1000BASE-X registers are enabled by writing to “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 1.

**Table 539: 1000BASE-X Auto-Negotiation Link Partner Ability Register—Base Page (Address 05h)
When in GMII/RGMII/RTBI Mode**

Bit	Name	R/W	Description	Default
15	Next Page	RO	1 = Link partner supports Next Page ability 0 = Link partner does not support Next Page ability	0
14	Acknowledge	RO	1 = Link partner has received link code word 0 = Link partner has not received link code word	0
13	Remote Fault	RO	00 = No remote fault	0
12		RO	01 = Link fault 10 = Off line 11 = Auto-negotiation error	0
11	Reserved	RO	Write as 0, ignore on read	0
10	Reserved	RO	Write as 0, ignore on read	0
9	Reserved	RO	Write as 0, ignore on read	0
8	Pause	RO	00 = Link partner sends no pause	0
7		RO	01 = Link partner sends symmetric pause 10 = Link partner sends asymmetric pause toward link partner 11 = Link partner sends both asymmetric and symmetric pause toward local device	0
6	Half-Duplex Capable	RO	1 = Link partner is 1000BASE-X half-duplex capable 0 = Link partner is not 1000BASE-X half-duplex capable	0
5	Full-Duplex Capable	RO	1 = Link partner is 1000BASE-X full-duplex capable 0 = Link partner is not 1000BASE-X full-duplex capable	0
4	Reserved	RO	Write as 0, ignore on read	0
3	Reserved	RO	Write as 0, ignore on read	0
2	Reserved	RO	Write as 0, ignore on read	0
1	Reserved	RO	Write as 0, ignore on read	0
0	Reserved	RO	Write as 0, ignore on read	0

Next Page

The NetXtreme II returns a 1 in bit 15 when the link partner wants to transmit Next Page information.

Acknowledge

The NetXtreme II returns a 1 in bit 14 when the link partner has acknowledged reception of the link code word; otherwise, it returns a 0.

Remote Fault

The NetXtreme II returns a nonzero value in bits[13:12] when the link partner has advertised detection of a remote fault; otherwise, it returns a 00.

Pause

The NetXtreme II returns values in bits[8:7] when the link partner has advertised pause capability.

Half-Duplex Capable

The NetXtreme II returns a 1 in bit 6 when the link partner has advertised 1000BASE-X half-duplex capability; otherwise, it returns a 0.

Full-Duplex Capable

The NetXtreme II returns a 1 in bit 5 when the link partner has advertised 1000BASE-X full-duplex capability; otherwise, it returns a 0.

SGMII AUTO-NEGOTIATION LINK PARTNER ABILITY REGISTER (ADDRESS 05H)

The following 1000BASE-X registers are enabled by writing to “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 1.

Table 540: 1000BASE-X Auto-Negotiation Link Partner Ability Register—Base Page (Address 05h) When in SGMII Mode

Bit	Name	R/W	Description	Default
15	Copper Link	RO	(Switch/MAC only sends 0)	0
14	Acknowledge	RO	1 = Link partner has received link code word 0 = Link partner has not received link code word	0
13	Reserved	RO	Ignore on read (Switch/MAC sends 0 only)	0
12	Copper Duplex	RO	(Switch/MAC only sends 0)	0
11	Copper Speed	RO	(Switch/MAC only sends 00)	0
10		RO		0
9	Reserved	RO	Ignore on read	0
8	Reserved	RO	Ignore on read	0
7	Reserved	RO	Ignore on read	0
6	Reserved	RO	Ignore on read	0
5	Reserved	RO	Ignore on read	0
4	Reserved	RO	Ignore on read	0
3	Reserved	RO	Ignore on read	0
2	Reserved	RO	Ignore on read	0
1	Reserved	RO	Ignore on read	0
0	SGMII Selector	RO	1 = SGMII mode	1

Note: As indicated by bit 5 of the 1000BASE-X MII Status register, the values contained in the Auto-Negotiation Link Partner Ability register are only guaranteed to be valid after auto-negotiation has successfully completed.

SGMII Mode

When SGMII mode is enabled, then reading the this register will reflect values sent from the LINK partner.

Copper Link

Bit15 indicates the link is established on the copper interface. The Switch/MAC sends 0 only.

Acknowledge

Bit14 indicates the SGMII link partner has received the link code word.

Copper Duplex

Bit12 indicates the copper link partner is linked up at full-duplex mode. The Switch/MAC sends 0 only.

Copper Speed

Bits[11:10] indicate the copper link-up speed. The Switch/MAC sends 00 only.



SGMII Selector

Bit 0 = 1 indicates the Switch/MAC is in SGMII mode.

1000BASE-X AUTO-NEGOTIATION EXTENDED STATUS REGISTER (ADDRESS 06H)

The following 1000BASE-X registers are enabled by writing to "[Mode Control Register \(Address 1Ch, Shadow 1Fh\)](#)" bit 0 = 1.

Table 541: 1000BASE-X Auto-Negotiation Extended Status Register (Address 06h)

Bit	Name	R/W	Description	Default
15	Reserved	RO	Write as 0, ignore on read	0
14	Reserved	RO	Write as 0, ignore on read	0
13	Reserved	RO	Write as 0, ignore on read	0
12	Reserved	RO	Write as 0, ignore on read	0
11	Reserved	RO	Write as 0, ignore on read	0
10	Reserved	RO	Write as 0, ignore on read	0
9	Reserved	RO	Write as 0, ignore on read	0
8	Reserved	RO	Write as 0, ignore on read	0
7	Reserved	RO	Write as 0, ignore on read	0
6	Reserved	RO	Write as 0, ignore on read	0
5	Reserved	RO	Write as 0, ignore on read	0
4	Reserved	RO	Write as 0, ignore on read	0
3	Reserved	RO	Write as 0, ignore on read	0
2	Next Page Capable	RO L	1 = Local device is next page capable 0 = Local device is not next page capable	0
1	Page Received	RO LH	1 = New page has been received from link partner 0 = New page has not been received	0
0	Reserved	RO	Ignore on read	0

Next Page Capable

Bit 2 is always set = 0, the NetXtreme II does not support Next Page capability and returns a 0.

Page Received

The NetXtreme II returns a 1 in bit 1 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

1000BASE-X IEEE EXTENDED STATUS REGISTER (ADDRESS 0Fh)

The following 1000BASE-X registers are enabled by writing to “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 1.

Table 542: 1000BASE-X IEEE Extended Status Register (Address 0Fh)

Bit	Name	R/W	Description	Default
15	1000BASE-X Full-Duplex Capable	RO H	1 = 1000BASE-X full-duplex capable 0 = Not 1000BASE-X full-duplex capable	1
14	1000BASE-X Half-Duplex Capable	RO H	1 = 1000BASE-X half-duplex capable 0 = Not 1000BASE-X half-duplex capable	1
13	1000BASE-T Full-Duplex Capable	RO L	1 = 1000BASE-T full-duplex capable 0 = Not 1000BASE-T full-duplex capable	0
12	1000BASE-T Half-Duplex Capable	RO L	1 = 1000BASE-T half-duplex capable 0 = Not 1000BASE-T half-duplex capable	0
11	Reserved	RO	Write as 0, ignore on read	0
10	Reserved	RO	Write as 0, ignore on read	0
9	Reserved	RO	Write as 0, ignore on read	0
8	Reserved	RO	Write as 0, ignore on read	0
7	Reserved	RO	Write as 0, ignore on read	0
6	Reserved	RO	Write as 0, ignore on read	0
5	Reserved	RO	Write as 0, ignore on read	0
4	Reserved	RO	Write as 0, ignore on read	0
3	Reserved	RO	Write as 0, ignore on read	0
2	Reserved	RO	Write as 0, ignore on read	0
1	Reserved	RO	Write as 0, ignore on read	0
0	Reserved	RO	Write as 0, ignore on read	0

1000BASE-X Full-Duplex Capable

The NetXtreme II is capable of 1000BASE-X full-duplex operation and returns a 1 when bit 15 is read.

1000BASE-X Half-Duplex Capable

The NetXtreme II is capable of 1000BASE-X half-duplex operation and returns a 1 when bit 14 is read.

1000BASE-T Full-Duplex Capable

The NetXtreme II is not capable of 1000BASE-T full-duplex operation and returns a 0 when bit 13r is read.

1000BASE-T Half-Duplex Capable

The NetXtreme II is not capable of 1000BASE-T half-duplex operation and returns a 0 when bit 12 is read.

EXPANSION REGISTERS

EXPANSION REGISTER 00H: RECEIVE/TRANSMIT PACKET COUNTER REGISTER (ADDRESS 15H)

Expansion register 00h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F00’h, and read/write access is through register 15h.

Table 543: Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)

Bit	Name	R/W	Description	Default
15:0	Packet Counter (Copper Only)	R/W CR	Returns the transmitted and received packet count.	0000h

Packet Counter (Copper Only)

The mode of this counter is set by bit 11 of “[1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register \(Address 18h, Shadow 7h\)](#)”. When bit 11 = 1, then receive packets (both good and bad CRC error packets) are counted. When bit 11 = 0, then transmit packets (both good and bad CRC error packets) are counted. This counter is cleared on read and freezes at FFFFh.

EXPANSION REGISTER 01H: EXPANSION INTERRUPT STATUS REGISTER (ADDRESS 15H)

Expansion register 02h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F02’h, and read/write access is through register 15h.

Table 544: Expansion Register 01h: Expansion Interrupt Status Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	RO	Write as 0, ignore on read	0
14	Reserved	RO	Write as 0, ignore on read	0
13	Reserved	RO	Write as 0, ignore on read	0
12	Reserved	RO	Write as 0, ignore on read	0
11	Reserved	RO	Write as 0, ignore on read	0
10	Reserved	RO	Write as 0, ignore on read	0
9	Reserved	RO	Write as 0, ignore on read	0
8	Reserved	RO	Write as 0, ignore on read	0
7	Mode Select Change	RO LH	1 = Mode select change detected (clears on read) 0 = Mode select change not detected	0
6	Fiber Link Status Change	RO LH	1 = Fiber link status change detected (clears on read) 0 = Mode select change not detected	0
5	1000BASE-X rudi_c detected	RO LH	1 = 1000BASE-X rudi_c detected (SerDes auto-negotiation code word received) since last read 0 = No 1000BASE-X rudi_c detected since last read	0

Table 544: Expansion Register 01h: Expansion Interrupt Status Register (Address 15h) (Cont.)

Bit	Name	R/W	Description	Default
4	Reserved	RO LH	Write as 0, ignore on read	0
3	Reserved	RO LH	Write as 0, ignore on read	0
2	Reserved	RO LH	Write as 0, ignore on read	0
1	Reserved	RO LH	Write as 0, ignore on read	0
0	Transmit CRC Error (Copper Only)	RO LH	1= Transmit CRC error detected since last read 0= No Transmit CRC error detected since last read	0

Mode Select Change

Bit 7 = 1 indicates that a mode select change is detected.

Fiber Link Status Change

Bit 6 = 1 indicates that a Fiber link status change is detected.

1000BASE-X rudi_C Detected

Bit 5 = 1 indicates that the Fiber auto-negotiation code word was received.

Transmit CRC Error (Copper Only)

Bit 0 = 1 indicates that a transmit CRC error has occurred since this register was last read. To enable the Transmit CRC checker, bit 3 in register 1Ch, shadow value 00010, must be set to a 1.

Expansion register 01h is enabled by writing to "[Expansion Register Access Register \(Address 17h\)](#)" bits 11:0 = 'F01'h, and read/write access is through register 15h.

EXPANSION REGISTER 02H: EXPANSION INTERRUPT MASK REGISTER (ADDRESS 15H)**Table 545: Expansion Register 02h: Expansion Interrupt Mask Register (Address 15h)**

Bit	Name	R/W	Description	Default
15	Reserved	RO	Write as 0 ignore on read	0
14	Reserved	RO	Write as 0 ignore on read	0
13	Reserved	RO	Write as 0 ignore on read.	0
12	Reserved	RO	Write as 0 ignore on read	0
11	Reserved	RO	Write as 0 ignore on read.	0
10	Reserved	RO	Write as 0 ignore on read	0
9	Reserved	RO	Write as 0 ignore on read	0
8	Reserved	RO	Write as 0 ignore on read	0
7	Mode Select Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
6	Fiber Link Status Change	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
5	1000BASE-X rudi_c detected	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1
4	Reserved	R/W	Write as 1, ignore on read	1
3	Reserved	R/W	Write as 1, ignore on read	1
2	Reserved	R/W	Write as 1, ignore on read	1
1	Reserved	R/W	Write as 1, ignore on read	1
0	Transmit CRC Error (Copper Only)	R/W	1 = Interrupt masked, status bits operate normally 0 = Interrupt enabled, status bits operate normally	1

Interrupt Mask Vector

When bit n of the Expansion Interrupt Mask register is written to 1, the interrupt corresponding to the same bit in the Expansion Interrupt Status register is masked. The status bits still operate normally when the interrupt is masked, but do not generate an interrupt output. When the bit is written to 0, the interrupt is unmasked

EXPANSION REGISTER 04H: MULTICOLOR LED SELECTOR REGISTER (ADDRESS 15H)

Expansion register 04h is enabled by writing to "[Expansion Register Access Register \(Address 17h\)](#)" bits 11:0 = 'F04'h, and read/write access is through register 15h.

Table 546: Expansion Register 04h: Multicolor LED Selector Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Reserved	R/W	Write as 0, ignore on read	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Flash Now SC	R/W	1 = Initiate a multicolor LED flash. This works only when the multicolor selector is set to 0111. 0 = MULTICOLOR[1] and MULTICOLOR[2] are in opposite phase. Note: This is only valid when Multicolor LED Selector bits are set to 0000, 0010, 0011, 0110, 0111, 1000, 1001, 1010.	0
8	In Phase	R/W	1 = MULTICOLOR[1] and MULTICOLOR[2] are in phase. 0 = MULTICOLOR[1] and MULTICOLOR[2] are in opposite phase. Note: This is only valid when Multicolor LED Selector bits are set to 0000, 0010, 0011, 0110, 0111, 1000, 1001, 1010.	0
7	<u>MULTICOLOR[2]</u> LED Selector	R/W	Selects the multicolor mode for MULTICOLOR[2] LED	0
6		R/W	0000: Encoded link/activity LED	0
5		R/W	0001: Encoded speed LED	0
4		R/W	0010: Activity flash LED 0011: Full-duplex LED 0100: Forced off 0101: Forced on 0110: Alternating LED (50% duty cycle with a 320 ms period) 0111: Flashing LED (toggling between 2 of the states with an 80 ms period) 1000: Link LED 1001: Activity LED 1010: Programmable blink LED	0
3	<u>MULTICOLOR[1]</u> LED Selector	R/W	Selects the multicolor mode for MULTICOLOR[1] LED	0
2		R/W	0000: Encoded link/activity LED	0
1		R/W	0001: Encoded speed LED	0
0		R/W	0010: Activity flash LED 0011: Full-duplex LED 0100: Forced off 0101: Forced on 0110: Alternating LED (50% duty cycle with a 320 ms period) 0111: Flashing LED (toggling between 2 of the states with an 80 ms period) 1000: Link LED 1001: Activity LED 1010: Programmable blink LED	0

Flash Now

Asserting this bit causes a single flash to occur on either MULTICOLOR[2:1] LEDs, as long as its multicolor selector is set to 0111.

In Phase

When both LEDs are selected to the same mode, the MULTICOLOR[2:1] output pins toggle at the same time. This bit determines whether the pins are identical to each other, or inverses of each other. When the two LED pins are attached to a special multicolored LED, the resulting LED colors alternate either between off/amber (in phase) or red/green (out of phase).

MULTICOLOR[2] LED Selector

The bits[7:4] select the multicolor LED mode for MULTICOLOR[2]. It is up to the user to determine what functions should appear on the two LED pins. For example, if the user wants a different color toggling operation rather than the operation mentioned above, such as red/amber, the user can put one of the selectors to the desired toggle mode, and other selector to *forced on*.

MULTICOLOR[1] LED Selector

Bits[3:0] select the multicolor LED mode for MULTICOLOR[1].

EXPANSION REGISTER 05H: MULTICOLOR LED FLASH RATE CONTROLS REGISTER (ADDRESS 15H)

Expansion register 05h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F05’h, and read/write access is through register 15h.

Table 547: Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Alternating Rate	R/W	Determines the width and gap for multicolor LED selector 0110 (alternating LED mode).	0
10		R/W	00h = 21 ms width, 21 ms gap	0
9		R/W	01h = 42 ms width, 42 ms gap	0
8		R/W	02h = 63 ms width, 63 ms gap	1
7		R/W	...	1
6		R/W	07h = 168 ms width, 168 ms gap	1
			...	
			3Fh = 1.344 seconds	
5	Flash Rate	R/W	Determines the width and minimum gap of every flash pulse for multicolor LED selector 0000 (encoded link/activity mode), 0010 (activity flash mode) and 0111 (flashing LED mode).	0
4		R/W	00h = 21 ms width	0
3		R/W	01h = 42 ms width	0
2		R/W	02h = 63 ms width	0
1		R/W	...	0
0		R/W	3Fh = 1.344 seconds	1

Alternating Rate

Setting Bits[11:6] changes the width and gap of the alternating LED modes. These bits are only valid when the MULTICOLOR[1] LED Selector and or the MULTICOLOR[2] LED Selector bits = 0110. The duty cycle of the LEDs is exactly 50%.

Flash Rate

Setting Bits[5:0] determines the width and minimum gap of the flashing pulse. These bits are only valid when the MULTICOLOR[1] LED Selector and or the MULTICOLOR[2] LED Selector bits = 0111. The duty cycle of the flash rate is not exactly 50%.

EXPANSION REGISTER 06H: MULTICOLOR LED PROGRAMMABLE BLINK CONTROLS REGISTER (ADDRESS 15H)

Expansion register 06h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F06’h, and read/write access is through register 15h.

Table 548: Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read	0
14	Reserved	R/W	Write as 0, ignore on read	0
13	Reserved	R/W	Write as 0, ignore on read	0
12	Reserved	R/W	Write as 0, ignore on read	0
11	Reserved	R/W	Write as 0, ignore on read	0
10	Reserved	R/W	Write as 0, ignore on read	0
9	Reserved	R/W	Write as 0, ignore on read	0
8	Reserved	R/W	Write as 0, ignore on read	0
7	Reserved	R/W	Write as 0, ignore on read	0
6	Reserved	R/W	Write as 0, ignore on read	0
5	Blink Update Now	R/W	1 = Change to the new blink rate now. 0 = Wait 1 second before changing the blink rate. Controls when a change in the blink rate is actually displayed on the Programmable Blink LED.	0
4	Blink Rate	R/W	Programs the number of blinks per second of the Programmable Blink LED	0
3		R/W	00000 = No blink	0
2		R/W	00001 = 1 blink per second	0
1		R/W	00010 = 2 blinks per second	0
0		R/W	00011 = 3 blinks per second	0
			...	
			11111 = 31 blinks per second	

Blink Update Now

Setting bit 5 = 1 updates the blink rate immediately. Clearing this bit causes the blink rate to be updated after the 1 second interval timer expires. This bit is only valid when the MULTICOLOR[1] LED Selector and or the MULTICOLOR[2] LED Selector bits = 1010.

Blink Rate

Setting bits[4:0] determines the blink rate of the Programmable Blink LED. These bits are only valid when the MULTICOLOR[1] LED Selector or the MULTICOLOR[2] LED Selector bits = 0000, 0010, or 0111.

Section 13: BCM5709 and BCM5716 Transceiver Register Summary

MII MANAGEMENT INTERFACE REGISTER PROGRAMMING

Access to the following transceiver registers is provided indirectly through the MII Communication Register (see “[EMAC MDIO Command Register \(emac_mdio_comm, Offset 0x14ac\)](#)” on page 321) of the BCM5709C and BCM5716 controllers using a PHY_ADDR field value of 1h. When the BCM5709S is used, these registers are only applicable when the device is operating in 1000BASE-T (Copper) mode.

REGISTER MAP

The NetXtreme II transceiver contains the following set of registers.

Table 549: Register Map

Address	Register
1000BASE-T/100BASE-TX/10BASE-T Registers	
00h	Table 550: “1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h),” on page 737
01h	Table 551: “1000BASE-T/100BASE-TX/10BASE-TMII Status Register (Address 01h),” on page 739
02h	Table 552: “1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register MSB (Address 02h),” on page 742
03h	Table 553: “1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register LSB (Address 03h),” on page 742
04h	Table 554: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h),” on page 743
05h	Table 555: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register (Address 05h),” on page 746
06h	Table 556: “1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register (Address 06h),” on page 749
07h	Table 557: “1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h),” on page 751
08h	Table 558: “1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h),” on page 752
09h	Table 559: “1000BASE-T Control Register (Address 09h),” on page 753
0Ah	Table 560: “1000BASE-T Status Register (Address 0Ah),” on page 755
0Fh	Table 561: “1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh),” on page 757
10h	Table 562: “1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h),” on page 758

Table 549: Register Map (Cont.)

Address	Register
11h	Table 563: “1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h),” on page 760
12h	Table 564: “1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h),” on page 763
13h	Table 565: “1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register (Address 13h),” on page 764
13h	Table 566: “1000BASE-T/100BASE-TX/10BASE-T Transmit Error Code Counter Register (Address 13h),” on page 765
14h	Table 567: “1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register (Address 14h),” on page 766
14h	Table 568: “1000BASE-T/100BASE-TX/10BASE-T CRC Counter Register (Address 14h),” on page 767
18h	Table 574: “1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow 0h),” on page 771 Table 575: “10BASE-T Register (Address 18h, Shadow 1h),” on page 773 Table 576: “1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow 2h),” on page 775 Table 577: “1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow 4h),” on page 777 Table 578: “1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow 7h),” on page 779
19h	Table 579: “1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h),” on page 781
1Ah	Table 580: “1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah),” on page 784
1Bh	Table 581: “1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh),” on page 787

Table 549: Register Map (Cont.)

Address	Register
1Ch	Table 586: “1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h),” on page 794 Table 585: “1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow 04h),” on page 792 Table 586: “1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h),” on page 794 Table 587: “1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow 08h),” on page 796 Table 588: “1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow 09h),” on page 798 Table 589: “Auto Power-Down Register (Address 1Ch, Shadow 0Ah),” on page 802 Table 590: “SLED_1 Register (Address 1Ch, Shadow 0Bh),” on page 804 Table 591: “SLED_2 Register (Address 1Ch, Shadow 0Ch),” on page 805 Table 592: “1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow 0Dh),” on page 806 Table 593: “1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow 0Eh),” on page 808 Table 594: “LED GPIO Control/Status Register (Address 1Ch, Shadow Value 0Fh),” on page 810 Table 596: “SGMII Status Register (Address 1Ch, Shadow 15h),” on page 812 Table 597: “Misc. 1000BASE-X Control 2 Register (Address 1Ch, Shadow 16h),” on page 814 Table 598: “1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h),” on page 816 Table 599: “Auxiliary Control Register (Address 1Ch, Shadow 1Bh),” on page 818 Table 602: “Mode Control Register (Address 1Ch, Shadow 1Fh),” on page 824
1Dh	Table 603: “1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0,” on page 826 Table 604: “1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1,” on page 828
1Eh	Table 605: “Test Register 1 (Address 1Eh),” on page 831
1Fh	Reserved (Do not read from or write to a reserved register.)
Expansion Registers: Read/Write through Register 15h (Accessed by Writing to Register 17h, Bits[11:0] = 1111 + Expansion Register Number)	
00h	Table 606: “Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h),” on page 833
04h	Table 607: “Expansion Register 04h: Multicolor LED Selector Register (Address 15h),” on page 834
05h	Table 608: “Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h),” on page 836
06h	Table 609: “Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h),” on page 837
07h	Table 610: “Expansion Register 07h: 100BASE-FX Far End Fault Register (Address 15h),” on page 838
0Bh	Table 612: “Expansion Register 0Bh: Port Interrupt Status Register (Address 15h),” on page 840
10h	Table 613: “Expansion Register 10h: Cable Diagnostic Controls and Basic Status Register (Address 15h),” on page 841
11h	Table 614: “Expansion Register 11h: Cable Diagnostic Results Register (Address 15h),” on page 842
12h	Table 615: “Expansion Register 12h: Cable Diagnostic Lengths Register (Address 15h),” on page 843
13h	Table 616: “Expansion Register 13h: Cable Diagnostic Lengths Register (Address 15h),” on page 843

Table 549: Register Map (Cont.)

Address	Register
42h	Table 617: "Expansion Register 42h: Operating Mode Status Register (Address 15h)," on page 844
44h	Table 618: "Expansion Register 44h: SGMII Lineside/Loopback Control Register (Address 15h)," on page 845
51h	Table 619: "Expansion Register 51h: SGOUT± Control Register (Address 15h)," on page 846
52h	Table 620: "Expansion Register 52h: SGOUT± and SCLK± Control Register (Address 15h)," on page 847
67h	Table 621: "Expansion Register 67h: 1000BASE-X Auto-Neg Misc. RX Status Register (Address 15h)," on page 848
70h	Table 622: "Expansion Register 70h: Soft Reset Register (Address 15h)," on page 849

REGISTER NOTATIONS

In the register description tables, the following notation in the R/W column is used to describe the ability to read or write:

- R/W = Read or write
- RO = Read only
- WO = Write only
- LH = Latched high (until read)
- LL = Latched low (until read)
- H = Fixed high
- L = Fixed low
- SC = Self-clear after read
- CR = Clear on read

Reserved bits must be written as the default value and ignored when read.

1000BASE-T/100BASE-TX/10BASE-T REGISTER DESCRIPTIONS

1000BASE-T/100BASE-TX/10BASE-T MII CONTROL REGISTER (ADDRESS 00H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 550: 1000BASE-T/100BASE-TX/10BASE-T MII Control Register (Address 00h)

Bit	Name	R/W	Description	Default
15	Reset	R/W SC	1 = PHY reset. 0 = Normal operation.	0
14	Internal Loopback	R/W	1 = Enable SGMII Loopback 0 = Normal operation.	0
13	Speed Selection (LSB)	R/W	Bits[6,13]: 1 1 = Reserved 1 0 = 1000 Mbps 0 1 = 100 Mbps 0 0 = 10 Mbps	0
12	Copper Auto-negotiation Enable	R/W	1 = Auto-negotiation enabled. 0 = Auto-negotiation disabled.	1
11	Power Down	R/W	1 = Power down. 0 = Normal operation.	0
10	Reserved	R/W	Write as 0, ignore on read.	0
9	Restart Auto-negotiation	R/W SC	1 = Restarting auto-negotiation. 0 = Auto-negotiation restart complete.	0
8	Duplex Mode	R/W	1 = Full-duplex. 0 = Half-duplex.	1
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Speed Selection (MSB)	R/W	Works in conjunction with bit 13.	1
5	Reserved	R/W	Write as 0, ignore on read.	0
4	Reserved	R/W	Write as 0, ignore on read.	0
3	Reserved	R/W	Write as 0, ignore on read.	0
2	Reserved	R/W	Write as 0, ignore on read.	0
1	Reserved	R/W	Write as 0, ignore on read.	0
0	Reserved	R/W	Write as 0, ignore on read.	0

Reset

To reset the NetXtreme II by software control, a 1 must be written to bit 15 of the MII Control register. This bit clears itself after the reset process is complete, and does not need to be cleared using a second MII write. Writes to other MII Control register bits have no effect until the reset process is completed, which requires approximately 2.0 μ s. Writing a 0 to this bit has no effect. A 1 is returned when this bit is read during the reset process; otherwise, it returns a 0.

Internal Loopback

When enabled packets received on the SGIN \pm pins will be loopback out the SGOUT \pm pins.

Speed Selection (LSB)

When auto-negotiation is disabled, bits 6 and 13 of the MII Control register can be used to manually select the speed of operation. When bit 6 is set and bit 13 is cleared, 1000BASE-T operation is selected. When bit 6 is cleared and bit 13 is set, 100BASE-TX operation is selected. When both bits are cleared, 10BASE-T operation is selected. Setting both bits is not permitted. When read, these bits return the last value written.

Copper Auto-Negotiation Enable

When bit 12 of the MII Control register is set, the NetXtreme II mode of operation is controlled by auto-negotiation. When this bit is cleared, the NetXtreme II mode of operation is determined by the Manual Speed, Duplex mode, and Master/Slave Configuration bits. A 1 is returned when this bit is read with auto-negotiation enabled; otherwise, it returns a 0.

Power-Down

When bit 11 of the MII Control register is set, the NetXtreme II is placed into low-power standby mode. When in this mode all NetXtreme II functions except for the MDIO/MDC, 1000BASE-X, and SGMII interfaces.

Restart Auto-Negotiation

Setting bit 9 of the MII Control register forces the auto-negotiation process to be restarted, regardless of the current state of the auto-negotiation state machine. When auto-negotiation is enabled, setting this bit restarts the auto-negotiation process. Clearing this bit has no effect. This bit is self-clearing. After the auto-negotiation process has restarted, it returns a value of 0.

Duplex Mode

When auto-negotiation is disabled, the duplex mode of the NetXtreme II can be controlled by writing to bit 8 of the MII Control register. Setting this bit forces the NetXtreme II into full-duplex operation, and clearing this bit forces the NetXtreme II into half-duplex operation.

Speed Selection (MSB)

When auto-negotiation is disabled, bits 6 and 13 of the MII Control register can be used to manually select the speed of operation. When bit 6 is set and bit 13 is cleared, 1000BASE-T operation is selected. When bit 6 is cleared and bit 13 is set, 100BASE-TX operation is selected. When both bits are cleared, 10BASE-T operation is selected. Setting both bits is not permitted. When read, these bits return the last value written.



1000BASE-T/100BASE-TX/10BASE-T MII STATUS REGISTER (ADDRESS 01H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 551: 1000BASE-T/100BASE-TX/10BASE-TMII Status Register (Address 01h)

Bit	Name	R/W	Description	Default
15	100BASE-T4 Capable	RO L	1 = 100BASE-T4 capable. 0 = Not 100BASE-T4 capable.	0
14	100BASE-X Full-Duplex Capable	RO H	1 = 100BASE-X full-duplex capable. 0 = Not 100BASE-X full-duplex capable.	1
13	100BASE-X Half-Duplex Capable	RO H	1 = 100BASE-X half-duplex capable. 0 = Not 100BASE-X half-duplex capable.	1
12	10BASE-T Full-Duplex Capable	RO H	1 = 10BASE-T full-duplex capable. 0 = Not 10BASE-T full-duplex capable.	1
11	10BASE-T Half-Duplex Capable	RO H	1 = 10BASE-T half-duplex capable. 0 = Not 10BASE-T half-duplex capable.	1
10	100BASE-T2 Full-Duplex Capable	RO L	1 = 100BASE-T2 full-duplex capable. 0 = Not 100BASE-T2 full-duplex capable.	0
9	100BASE-T2 Half-Duplex Capable	RO L	1 = 100BASE-T2 half-duplex capable. 0 = Not 100BASE-T2 half-duplex capable.	0
8	Extended Status	RO H	1 = Extended status information in register 0Fh. 0 = No extended status information in register 0Fh.	1
7	Reserved	RO	Ignore on read.	0
6	Management Frames Preamble Suppression	RO H	1 = Preamble can be suppressed. 0 = Preamble always required.	1
5	Auto-negotiation Complete	RO	1 = Auto-negotiation complete. 0 = Auto-negotiation in progress.	0
4	Remote Fault	RO LH	1 = Remote fault detected. 0 = No remote fault detected.	0
3	Auto-negotiation Ability	RO H	1 = Auto-negotiation capable. 0 = Not auto-negotiation capable.	1
2	Copper Link Status	RO LL	1 = Link is up (Link Pass state). 0 = Link is down (Link Fail state).	0
1	Jabber Detect	RO LH	1 = Jabber condition detected. 0 = No jabber condition detected.	0
0	Extended Capability	RO H	1 = Extended register capabilities. 0 = No extended register capabilities.	1

100BASE-T4 Capable

The NetXtreme II is not capable of 100BASE-T4 operation and returns a 0 when bit 15 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

100BASE-X Full-Duplex Capable

The NetXtreme II is capable of 100BASE-TX full-duplex operation and returns a 1 when bit 14 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

100BASE-X Half-Duplex Capable

The NetXtreme II is capable of 100BASE-X half-duplex operation and returns a 1 when bit 13 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

10BASE-T Full-Duplex Capable

The NetXtreme II is capable of 10BASE-T full-duplex operation and returns a 1 when bit 12 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

10BASE-T Half-Duplex Capable

The NetXtreme II is capable of 10BASE-T half-duplex operation and returns a 1 when bit 11 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register is read.

100BASE-T2 Full-Duplex Capable

The NetXtreme II is not capable of 100BASE-T2 full-duplex operation and returns a 0 when bit 10 is read.

100BASE-T2 Half-Duplex Capable

The NetXtreme II is not capable of 100BASE-T2 half-duplex operation and returns a 0 when bit 9 is read.

Extended Status

The NetXtreme II contains IEEE Extended Status register at address 0Fh and returns a 1 when bit 8 is read.

Management Frames Preamble Suppression

The NetXtreme II accepts MII management frames whether or not they are preceded by the preamble pattern, and returns a 1 when bit 6 is read.



Note: Preamble is still required on the first read or write. There is no way to disable Preamble Suppression.

Auto-Negotiation Complete

The NetXtreme II returns a 1 in bit 5, and the contents of registers 4, 5, and 6 are valid. This bit returns a 0 while auto-negotiation is in progress.

Remote Fault

The NetXtreme II returns a 1 in bit 4 when its link partner has signaled a remote fault condition. When a remote fault occurs, the bit is set and remains so until the remote fault condition has been cleared and the register is read.

Auto-Negotiation Ability

Even if the auto-negotiation function has been disabled, the NetXtreme II is capable of performing IEEE auto-negotiation and returns a 1 when bit 3 is read.

Copper Link Status

The NetXtreme II returns a 1 in bit 2 when the link monitor is in the link pass state (indicating that a 10BASE-T, 100BASE-TX or 1000BASE-X valid link has been established), otherwise, it returns a 0. When a link failure occurs, the Link Status bit is latched at 0 and remains so until the bit is read, and the NetXtreme II is in the link pass state.

Jabber Detect

Jabber detection is performed within the PHY and the result is latched into this bit. When a jabber condition has been detected, the NetXtreme II returns a 1 in bit 1. The bit is cleared by reading.

Extended Capability

The NetXtreme II supports Extended Capability registers and returns a 1 when bit 0 is read.

1000BASE-T/100BASE-TX/10BASE-T PHY IDENTIFIER REGISTER (ADDRESS 02H AND 03H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 552: 1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register MSB (Address 02h)

Bit	Name	R/W	Description	Default
15:0	OUI	RO	Bits[3:18] of Organizationally Unique Identifier.	0143 (hex)

Table 553: 1000BASE-T/100BASE-TX/10BASE-T PHY Identifier Register LSB (Address 03h)

Bit	Name	R/W	Description	Default
15:10	OUI	RO	Bits[19:24] of Organizationally Unique Identifier.	101111
9:4	MODEL	RO	Device model number.	111100
3:0	REVISION	RO	Device revision number.	n^a (hex)

^a = The revision number (n) changes with each silicon revision.

The IEEE has issued an Organizationally Unique Identifier (OUI) to Broadcom Corporation. This 24-bit number allows devices made by Broadcom to be distinguished from all other manufacturers. The OUI combined with model numbers and revision numbers assigned by Broadcom precisely identifies a device manufactured by Broadcom.

Bits[15:0] of MII register 02h (PHYID HIGH) contain OUI bits[3:18]. Bits[15:0] of MII register 03h (PHYID LOW) contain the most significant OUI bits[19:24]; six manufacturer model number bits, and four revision number bits. The two least significant OUI binary bits are not used.

Broadcom Corporation's OUI is 00-0A-F7 (expressed as hexadecimal values). The binary OUI is 0000-0000-0101-0000-1110-1111. The model number for the NetXtreme II is 2Ch. Revision numbers start with 0h and increment by 1 for each chip modification.

- PHYID HIGH[15:0] = OUI[3:18]
- PHYID LOW[15:0] = OUI[19:24] + Model[5:0] + Revision [3:0]

1000BASE-T/100BASE-TX/10BASE-T AUTO-NEGOTIATION ADVERTISEMENT REGISTER (ADDRESS 04H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 554: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Advertisement Register (Address 04h)

Bit	Name	R/W	Description	Default
15	Next Page	R/W	1 = Next page ability supported. 0 = Next page ability not supported.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Remote Fault	R/W	1 = Advertise remote fault detected. 0 = Advertise no remote fault detected.	0
12	Reserved Technology	R/W	Write as 0, ignore on read.	0
11	Asymmetric Pause	R/W	Used in conjunction with bit 10 to advertise the switch's pause capabilities. <u>Bit 10</u> <u>Bit 11</u> <u>PAUSE ABILITY</u> 0 0 No Pause 0 1 Asymmetric PAUSE toward link partner 1 0 Symmetric PAUSE 1 1 Both Symmetric PAUSE and Asymmetric PAUSE toward local device	0
10	Pause Capable	R/W	Used in conjunction with bit 11 to advertise the switch's pause capabilities.	0
9	100BASE-T4 Capable	R/W	1 = 100BASE-T4 capable. 0 = Not 100BASE-T4 capable.	0
8	100BASE-TX Full-Duplex Capable	R/W	1 = 100BASE-TX full-duplex capable. 0 = Not 100BASE-TX full-duplex capable.	1
7	100BASE-TX Half-Duplex Capable	R/W	1 = 100BASE-TX half-duplex capable. 0 = Not 100BASE-TX half-duplex capable.	1
6	10BASE-T Full-Duplex Capable	R/W	1 = 10BASE-T full-duplex capable. 0 = Not 10BASE-T full-duplex capable.	1
5	10BASE-T Half-Duplex Capable	R/W	1 = 10BASE-T half-duplex capable. 0 = Not 10BASE-T half-duplex capable.	1
4	Protocol Selector Field	R/W	Bits[4:0] = 00001 indicates IEEE 802.3 CSMA/CD	0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		1

Next Page

Bit 15 must be set = 1 when the management software wants to control Next Page exchange. When this bit is cleared, Next Page exchange is automatically controlled by the NetXtreme II. When this bit is cleared and the NetXtreme II is not advertising 1000BASE-T capability, no Next Page exchange occurs.

Remote Fault

Setting bit 13 sends a remote fault indication to the link partner during auto-negotiation. Writing a 0 to this bit clears the Remote Fault transmission bit. This bit returns a 1 when advertising remote fault; otherwise, it returns a 0.

Reserved Technology

Bit 12 is reserved for future versions of the auto-negotiation standard and must always be written as 0.

Asymmetric Pause

Bit 11 and bit 10 are used to advertise the Pause capability of the switch per the IEEE 802.3 specification.

Bit 10 Bit 11 PAUSE ABILITY

0	0	No Pause
0	1	Asymmetric PAUSE toward link partner
1	0	Symmetric PAUSE
1	1	Both Symmetric PAUSE and Asymmetric PAUSE toward local device

Pause Capable

Bit 10 is used in conjunction with bit 11 to advertise the Pause capability of the switch.

100BASE-T4 Capable

The NetXtreme II does not support 100BASE-T4 capability. Do not write a 1 to bit 9.

100BASE-TX Full-Duplex Capable

When bit 8 is set = 1, the NetXtreme II advertises 100BASE-TX full-duplex capability. When bit 8 is cleared, the NetXtreme II advertises no 100BASE-TX full-duplex capability. This bit returns a 1 when advertising 100BASE-TX full-duplex capability; otherwise, it returns a 0.

100BASE-TX Half-Duplex Capable

When bit 7 is set = 1, the NetXtreme II advertises 100BASE-TX half-duplex capability. When bit 7 is cleared, the NetXtreme II advertises no 100BASE-TX half-duplex capability. This bit returns a 1 when advertising 100BASE-TX half-duplex capability; otherwise, it returns a 0.

10BASE-T Full-Duplex Capable

When bit 6 is set = 1, the NetXtreme II advertises 10BASE-T full-duplex capability. When this bit is cleared, the NetXtreme II advertises no 10BASE-T full-duplex capability. This bit returns a 1 when advertising 10BASE-T full-duplex capability; otherwise, it returns a 0.

10BASE-T Half-Duplex Capable

When bit 5 is set = 1, the NetXtreme II advertises 10BASE-T half-duplex capability. When bit 5 is cleared, the NetXtreme II advertises no 10BASE-T half-duplex capability. This bit returns a 1 when advertising 10BASE-T half-duplex capability; otherwise, it returns a 0.

Selector Field

Bits[4:0] indicate the protocol type. The value 00001 indicates that the NetXtreme II belongs to the 802.3 class of PHY transceivers.

1000BASE-T/100BASE-TX/10BASE-T AUTO-NEGOTIATION LINK PARTNER ABILITY REGISTER (ADDRESS 05H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 555: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Link Partner Ability Register (Address 05h)

Bit	Name	R/W	Description	Default															
15	Next Page	RO	1 = Link partner has Next Page ability. 0 = Link partner does not have Next Page ability.	0															
14	Acknowledge	RO	1 = Link partner has received link code word. 0 = Link partner has not received link code word.	0															
13	Remote Fault	RO	1 = Link partner has detected remote fault. 0 = Link partner has not detected remote fault.	0															
12	Reserved Technology	RO	Write as 0, ignore on read.	0															
11	Link Partner Asymmetric Pause	RO	Used in conjunction with bit 10 to determine the link partner's pause capabilities. <table> <thead> <tr> <th>Bit 10</th> <th>Bit 11</th> <th>PAUSE ABILITY</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No Pause capability</td> </tr> <tr> <td>0</td> <td>1</td> <td>Asymmetric PAUSE toward link partner</td> </tr> <tr> <td>1</td> <td>0</td> <td>Symmetric PAUSE</td> </tr> <tr> <td>1</td> <td>1</td> <td>Both Symmetric PAUSE and Asymmetric PAUSE toward local device</td> </tr> </tbody> </table>	Bit 10	Bit 11	PAUSE ABILITY	0	0	No Pause capability	0	1	Asymmetric PAUSE toward link partner	1	0	Symmetric PAUSE	1	1	Both Symmetric PAUSE and Asymmetric PAUSE toward local device	0
Bit 10	Bit 11	PAUSE ABILITY																	
0	0	No Pause capability																	
0	1	Asymmetric PAUSE toward link partner																	
1	0	Symmetric PAUSE																	
1	1	Both Symmetric PAUSE and Asymmetric PAUSE toward local device																	
10	Pause Capable	RO	Used in conjunction with bit 11 to determine the link partner's pause capabilities.	0															
9	100BASE-T4 Capable	RO	1 = Link partner is 100BASE-T4 capable. 0 = Link partner is not 100BASE-T4 capable.	0															
8	100BASE-TX Full-Duplex Capable	RO	1 = Link partner is 100BASE-TX full-duplex capable. 0 = Link partner is not 100BASE-TX full-duplex capable.	0															
7	100BASE-TX Half-Duplex Capable	RO	1 = Link partner is 100BASE-TX half-duplex capable. 0 = Link partner not 100BASE-TX half-duplex capable.	0															
6	10BASE-T Full-Duplex Capable	RO	1 = Link partner is 10BASE-T full-duplex capable. 0 = Link partner is not 10BASE-T full-duplex capable.	0															
5	10BASE-T Half-Duplex Capable	RO	1 = Link partner is 10BASE-T half-duplex capable. 0 = Link partner is not 10BASE-T half-duplex capable.	0															
4	Protocol Selector Field	RO	Link partner protocol selector field.	0															
3		RO		0															
2		RO		0															
1		RO		0															
0		RO		0															



Note: As indicated by bit 5 of the 1000BASE-T/100BASE-TX/10BASE-T MII Status register, the values contained in the 1000BASE-T/100BASE-TX/10BASE-T Auto-negotiation Link Partner Ability register are only guaranteed to be valid after auto-negotiation has successfully completed.

Next Page

The NetXtreme II returns a 1 in bit 15 when the link partner wants to transmit Next Page information.

Acknowledge

The NetXtreme II returns a 1 in bit 14 when the link partner has acknowledged reception of the link code word; otherwise, it returns a 0.

Remote Fault

The NetXtreme II returns a 1 in bit 13 when the link partner has advertised detection of a remote fault; otherwise, it returns a 0.

Reserved Technology

Bit 12 is reserved for future versions of the auto-negotiation standard and must be ignored when read.

Asymmetric Pause

Bit 11 and bit 10 are used to determine the link partner's Pause capability per the IEEE 802.3 specification.

<u>Bit 10</u>	<u>Bit 11</u>	<u>PAUSE ABILITY</u>
0	0	Link Partner has no Pause capability
0	1	Link Partner has Asymmetric PAUSE capability
1	0	Link Partner has Symmetric PAUSE capability
1	1	Link Partner has both Symmetric PAUSE and Asymmetric PAUSE capability

Pause Capable

The NetXtreme II returns a 1 in bit 10 when the link partner has advertised Pause Capability; otherwise, it returns a 0.

100BASE-T4 Capable

The NetXtreme II returns a 1 in bit 9 when the link partner has advertised 100BASE-T4 capability; otherwise, it returns a 0.

100BASE-TX Full-Duplex Capable

The NetXtreme II returns a 1 in bit 8 when the link partner has advertised 100BASE-TX full-duplex capability; otherwise, it returns a 0.

100BASE-TX Half-Duplex Capable

The NetXtreme II returns a 1 in bit 7 when the link partner has advertised 100BASE-TX half-duplex capability; otherwise, it returns a 0.

10BASE-T Full-Duplex Capable

The NetXtreme II returns a 1 in bit 6 when the link partner has advertised 10BASE-T full-duplex capability; otherwise, it returns a 0.

10BASE-T Half-Duplex Capable

The NetXtreme II returns a 1 in bit 5 when the link partner has advertised 10BASE-T half-duplex capability; otherwise, it returns a 0.

Protocol Selector Field

Bits[4:0] return the value of the link partner advertised Protocol Selector field.

1000BASE-T/100BASE-TX/10BASE-T AUTO-NEGOTIATION EXPANSION REGISTER (ADDRESS 06H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

**Table 556: 1000BASE-T/100BASE-TX/10BASE-T Auto-Negotiation Expansion Register
(Address 06h)**

Bit	Name	R/W	Description	Default
15	Reserved	RO	Ignore on read.	0
14	Reserved	RO	Ignore on read.	0
13	Reserved	RO	Ignore on read.	0
12	Reserved	RO	Ignore on read.	0
11	Reserved	RO	Ignore on read.	0
10	Reserved	RO	Ignore on read.	0
9	Reserved	RO	Ignore on read.	0
8	Reserved	RO	Ignore on read.	0
7	Reserved	RO	Ignore on read.	0
6	Next Page Receive Location Able	R/W	1 = Bit 5 in register 06h determines Next Page receive location. 0 = Bit 5 in register 06h does not determine Next Page receive location.	1
5	Next Page Receive Location	R/W	1 = Next Pages stored in register 08h. 0 = Next Pages stored in register 05h.	1
4	Parallel Detection Fault	RO LH	1 = Parallel detection fault detected. 0 = Parallel detection fault not detected.	0
3	Link Partner Next Page Ability	RO	1 = Link partner has Next Page capability. 0 = Link partner does not have Next Page capability.	0
2	Next Page Capable	RO H	1 = NetXtreme II is Next Page capable. 0 = NetXtreme II is not Next Page capable.	1
1	Page Received	RO LH	1 = New page has been received from link partner. 0 = New page has not been received.	0
0	Link Partner Auto-negotiation Ability	RO	1 = Link partner has auto-negotiation capability. 0 = Link partner does not have auto-negotiation.	0

Next Page Receive Location Able

When bit 6 = 1, the Next Page receive location is determined by bit 5.

Next Page Receive Location

When bit 5 = 1, the Next Pages are stored in register 05h. When bit 5 = 0, the Next Pages are stored in register 08h.

Parallel Detection Fault

When a parallel detection fault has occurred in the auto-negotiation state machine, bit 4 returns a 1. When a parallel detection fault occurs, this bit is latched at 1 and remains so until the register is read. If a parallel detection fault has not occurred since the last time it was read, this bit returns a 0.

Link Partner Next Page Ability

The NetXtreme II returns a 1 in bit 3 when the link partner needs to transmit Next Page information; otherwise, it returns a 0. This bit is a copy of bit 15 of the 1000BASE-T/100BASE-TX/10BASE-T Link Partner Ability register.

Next Page Capable

When bit 2 is read, the NetXtreme II supports Next Page capability and returns a 1.

Page Received

The NetXtreme II returns a 1 in bit 1 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

Link Partner Auto-Negotiation Ability

When the link partner is known to have auto-negotiation capability, the NetXtreme II returns a 1 in bit 0. Before any auto-negotiation information is exchanged or if the link partner does not comply with IEEE auto-negotiation, the bit returns a 0.

1000BASE-T/100BASE-TX/10BASE-T NEXT PAGE TRANSMIT REGISTER (ADDRESS 07H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 557: 1000BASE-T/100BASE-TX/10BASE-T Next Page Transmit Register (Address 07h)

Bit	Name	R/W	Description	Default
15	Next Page	R/W	1 = Additional Next Pages follow. 0 = Sending last Next Page.	0
14	Reserved	RO	Ignore on read	0
13	Message Page	R/W	1 = Formatted page. 0 = Unformatted page.	1
12	Acknowledge2	R/W	1 = Complies with message. 0 = Cannot comply with message. Not used with 1000BASE-T Next Pages.	0
11	Toggle	RO	Toggles between exchanges of different Next Pages.	0
10:0	Message/Unformatted Code Field	R/W	Next Page message code or unformatted data.	0

Next Page

Bit 15 must be set = 1 to indicate that more Next Pages are to be sent. This bit must be cleared to indicate that this is the last Next Page to be transmitted. When this bit is read, it returns the last value written.

Message Page

Bit 13 must be set = 1 to indicate that a formatted message page is being sent. This bit must be cleared to indicate that an unformatted page is being sent. When this bit is read, it returns the last value written.

Acknowledge2

When this bit is set = 1, the NetXtreme II indicates that it can comply with the Next Page request. When this bit is cleared, the NetXtreme II indicates that it cannot comply with the Next Page request. When this bit is read, it returns the last value written.

Toggle

This bit toggles between different Next Page exchanges to ensure a functional synchronization to the link partner.

Message/Unformatted Code Field

These 11 bits make up the message code defined by IEEE Std 802.3, Clause 28, Annex C, when sending formatted pages. When sending unformatted Next Pages, these 11 bits contain an arbitrary data value.

1000BASE-T/100BASE-TX/10BASE-T LINK PARTNER RECEIVED NEXT PAGE REGISTER (ADDRESS 08H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 558: 1000BASE-T/100BASE-TX/10BASE-T Link Partner Received Next Page Register (Address 08h)

Bit	Name	R/W	Description	Default
15	Next Page	RO	1 = Additional next pages follow. 0 = Sending last Next Page.	0
14	Acknowledge	RO	1 = Acknowledge 0 = No acknowledge.	0
13	Message Page	RO	1 = Formatted page. 0 = Unformatted page.	0
12	Acknowledge2	RO	1 = Complies with message. 0 = Cannot comply with message. Not used with 1000BASE-T Next Pages.	0
11	Toggle	RO	Toggles between exchanges of different Next Pages.	0
10:0	Message Code field	RO	Next Page message code or unformatted data.	0

Next Page

When the link partner has indicated that more Next Pages are to be sent, bit 15 returns a 1. This bit returns a 0 when the link partner has indicated that this is the last Next Page to be transmitted.

Acknowledge

Bit 14 returns a 1 to indicate that the link partner has received and acknowledged a Next Page. The bit returns a 0 until the link partner has acknowledged the page.

Message Page

Bit 13 returns a 1 to indicate that the link partner has sent a formatted message page. This bit returns a 0 when the link partner has sent an unformatted page.

Acknowledge 2

When the link partner has indicated that it can comply with the Next Page request, bit 12 returns a 1. When the link partner has indicated that it cannot comply with the Next Page request, this bit returns a 0.

Toggle

To ensure a functional synchronization to the NetXtreme II transceiver, the link partner toggles this bit between different Next Page exchanges.

Message Code Field

These 11 bits make up the message code defined by IEEE Std 802.3, Clause 28, Annex C, when the link partner has sent a formatted page. When the link partner has sent unformatted Next Pages, these 11 bits contain an arbitrary data value.

1000BASE-T CONTROL REGISTER (ADDRESS 09H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

**Table 559: 1000BASE-T Control Register
(Address 09h)**

Bit	Name	R/W	Description	Default
15	Test Mode	R/W	1 X X = Test mode 4—transmitter distortion test	0
14		R/W	0 1 1 = Test mode 3—slave transmit jitter test	0
13		R/W	0 1 0 = Test mode 2—master transmit jitter test	0
			0 0 1 = Test mode 1—transmit waveform test	
			0 0 0 = Normal operation.	
12	Master/Slave Configuration Enable	R/W	1 = Enable master/slave manual configuration value. 0 = Automatic master/slave configuration.	0
11	Master/Slave Configuration Value	R/W	1 = Configure PHY as master. 0 = Configure PHY as slave.	0
10	Repeater/DTE	R/W	1 = Repeater/switch device port. 0 = DTE device.	0
9	Advertise 1000BASE-T Full-Duplex Capability	R/W	1 = Advertise 1000BASE-T full-duplex capability. 0 = Advertise no 1000BASE-T full-duplex capability.	1
8	Advertise 1000BASE-T Half-Duplex Capability	R/W	1 = Advertise 1000BASE-T half-duplex capability. 0 = Advertise no 1000BASE-T half-duplex capability.	1
7	Reserved	RO	Ignore on read.	0
6	Reserved	RO	Ignore on read.	0
5	Reserved	RO	Ignore on read.	0
4	Reserved	RO	Ignore on read.	0
3	Reserved	RO	Ignore on read.	0
2	Reserved	RO	Ignore on read.	0
1	Reserved	RO	Ignore on read.	0
0	Reserved	RO	Ignore on read.	0

Test Mode

The NetXtreme II can be placed in one of four transmit test modes by writing bits[15:13] of the 1000BASE-T Control register. The transmit test modes are defined in IEEE 802.3ab. When read, these bits return the last value written. For test modes 1, 2, and 4, the PHY must have auto-negotiation disabled, forced to 1000BASE-T Master mode, and Auto-MDIX disabled.

- Disable auto-negotiation and force to 1000BASE-T mode (write to register 00h = 0x0140)
- Disable Auto-MDIX (write to register 18h, shadow 7h, bit 9 = 0)
- Force to Master Mode (write to register 09h, bits[12:11] = 11)
- Enter test mode (write to register 09h, bits[15:13] = the test mode you want)

Master/Slave Configuration Enable

When bit 12 is set = 1, the NetXtreme II master/slave mode is configured using the manual master/slave configuration value. When this bit is cleared, the master/slave mode is configured using the automatic resolution function. This bit returns a 1 when manual master/slave configuration is enabled; otherwise, it returns a 0.

Master/Slave Configuration Value

When bit 12 is set = 1, bit 11 determines the NetXtreme II master/slave mode of operation. When bit 11 is set, the NetXtreme II is configured as the master. When bit 11 is cleared, the NetXtreme II is configured as the slave. When read, this bit returns the last value written.

Repeater/DTE

When bit 10 is set = 1, the NetXtreme II advertises that it is a repeater or switch device port. When this bit is cleared, the NetXtreme II advertises that it is a DTE port. The advertised value is used in the automatic master/slave configuration resolution. The link partner which advertises repeater mode is configured to master if the opposing link partner advertises data terminal equipment (DTE); otherwise, this bit has no effect. This bit returns a 1 when advertising repeater/switch mode; otherwise, it returns a 0.

Advertise 1000BASE-T Full-Duplex Capability

When bit 9 is set = 1, the NetXtreme II advertises 1000BASE-T full-duplex capability. When this bit is cleared, the NetXtreme II advertises no 1000BASE-T full-duplex capability. This bit returns a 1 when advertising 1000BASE-T full-duplex capability; otherwise, it returns a 0.

Advertise 1000BASE-T Half-Duplex Capability

When bit 8 is set = 1, the NetXtreme II advertises 1000BASE-T half-duplex capability. When this bit is cleared, the NetXtreme II advertises no 1000BASE-T half-duplex capability. This bit returns a 1 when advertising 1000BASE-T half-duplex capability; otherwise, it returns a 0.

1000BASE-T STATUS REGISTER (ADDRESS 0Ah)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

**Table 560: 1000BASE-T Status Register
(Address 0Ah)**

Bit	Name	R/W	Description	Default
15	Master/Slave Configuration Fault	RO	1 = Master/slave configuration fault detected. LH 0 = No master/slave configuration fault detected.	0
14	Master/Slave Configuration Resolution	RO	1 = Local transmitter is master. 0 = Local transmitter is slave.	0
13	Local Receiver Status	RO	1 = Local receiver status is good. 0 = Local receiver status is not good.	0
12	Remote Receiver Status	RO	1 = Remote receiver status is good. 0 = Remote receiver status is not good.	0
11	Link Partner 1000BASE-T Full-Duplex Capability	RO	1 = Link partner is 1000BASE-T full-duplex capable. 0 = Link partner not 1000BASE-T full-duplex capable.	0
10	Link Partner 1000BASE-T Half-Duplex Capability	RO	1 = Link partner is 1000BASE-T half-duplex capable. 0 = Link partner not 1000BASE-T half-duplex capable.	0
9	Reserved	RO	Ignore on read.	0
8	Reserved	RO	Ignore on read.	0
7	Idle Error Count	RO	Number of idle errors since last read. CR	0
6	Idle Error Count	RO	Number of idle errors since last read. CR	0
5	Idle Error Count	RO	Number of idle errors since last read. CR	0
4	Idle Error Count	RO	Number of idle errors since last read. CR	0
3	Idle Error Count	RO	Number of idle errors since last read. CR	0
2	Idle Error Count	RO	Number of idle errors since last read. CR	0
1	Idle Error Count	RO	Number of idle errors since last read. CR	0
0	Idle Error Count	RO	Number of idle errors since last read. CR	0



Note: As indicated by bit 5 of the MII Status register (0h), the values contained in bits 14, 11, and 10 of the 1000BASE-T Status register are guaranteed to be valid only after auto-negotiation has successfully completed.

Master/Slave Configuration Fault

When a master/slave configuration fault occurs during auto-negotiation, the NetXtreme II returns a 1 in bit 15. When a configuration fault occurs, the bit is latched at 1 and remains so until either the register is read, auto-negotiation is restarted by writing bit 9 in the MII Control register, or auto-negotiation completes successfully with no master/slave configuration fault.

Master/Slave Configuration Resolution

When the NetXtreme II transceiver has been configured as the master, it returns a 1 in bit 14. When the NetXtreme II transceiver has been configured as the slave, it returns a 0.

Local Receiver Status

The NetXtreme II transceiver returns a 1 in bit 13 when the local receiver status is good; otherwise, it returns a 0.

Remote Receiver Status

The NetXtreme II returns a 1 in bit 12 when the remote receiver status is good; otherwise, it returns a 0.

1000BASE-T Full-Duplex Capability

The NetXtreme II returns a 1 in bit 11 when the link partner has advertised 1000BASE-T full-duplex capability; otherwise, it returns a 0.

1000BASE-T Half-Duplex Capability

The NetXtreme II returns a 1 in bit 10 when the link partner has advertised 1000BASE-T half-duplex capability; otherwise, it returns a 0.

Idle Error Count

The NetXtreme II counts the number of idle errors received while the local receiver status is good. Bits 7 through 0 return the number of idle errors counted since the last time the register was read. The counter freezes at the maximum value (FFh) to prevent overflow.

1000BASE-T/100BASE-TX/10BASE-T IEEE EXTENDED STATUS REGISTER (ADDRESS 0Fh)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 561: 1000BASE-T/100BASE-TX/10BASE-T IEEE Extended Status Register (Address 0Fh)

Bit	Name	R/W	Description	Default
15	1000BASE-X Full-Duplex Capable	RO L	1 = 1000BASE-X full-duplex capable. 0 = Not 1000BASE-X full-duplex capable.	0
14	1000BASE-X Half-Duplex Capable	RO L	1 = 1000BASE-X half-duplex capable. 0 = Not 1000BASE-X half-duplex capable.	0
13	1000BASE-T Full-Duplex Capable	RO H	1 = 1000BASE-T full-duplex capable. 0 = Not 1000BASE-T full-duplex capable.	1
12	1000BASE-T Half-Duplex Capable	RO H	1 = 1000BASE-T half-duplex capable. 0 = Not 1000BASE-T half-duplex capable.	1
11	Reserved	RO	Ignore on read.	0
10	Reserved	RO	Ignore on read.	0
9	Reserved	RO	Ignore on read.	0
8	Reserved	RO	Ignore on read.	0
7	Reserved	RO	Ignore on read.	0
6	Reserved	RO	Ignore on read.	0
5	Reserved	RO	Ignore on read.	0
4	Reserved	RO	Ignore on read.	0
3	Reserved	RO	Ignore on read.	0
2	Reserved	RO	Ignore on read.	0
1	Reserved	RO	Ignore on read.	0
0	Reserved	RO	Ignore on read.	0

1000BASE-X Full-Duplex Capable

The NetXtreme II is not capable of 1000BASE-X full-duplex operation and returns a 0 when bit 15 is read.

1000BASE-X Half-Duplex Capable

The NetXtreme II is not capable of 1000BASE-X half-duplex operation and returns a 0 when bit 14 is read.

1000BASE-T Full-Duplex Capable

The NetXtreme II is capable of 1000BASE-T full-duplex operation and returns a 1 when bit 13 is read.

1000BASE-T Half-Duplex Capable

The NetXtreme II is capable of 1000BASE-T half-duplex operation and returns a 1 when bit 12 is read.

1000BASE-T/100BASE-TX/10BASE-T PHY EXTENDED CONTROL REGISTER (ADDRESS 10H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 562: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Control Register (Address 10h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Disable Automatic MDI Crossover	R/W	1 = Automatic MDI crossover disabled. 0 = Automatic MDI crossover enabled.	0
13	Transmit Disable	R/W	1 = Transmitter outputs disabled. 0 = Normal operation.	0
12	Interrupt Disable	R/W	1 = Interrupt status output disabled. 0 = Interrupt status output enabled.	0
11	Force Interrupt	R/W	1 = Force interrupt status to active. 0 = Normal operation.	0
10	Bypass 4B/5B Encoder/Decoder (100BASE-TX)	R/W	1 = Transmit and receive 5B codes over MII pins. 0 = Normal MII.	0
9	Bypass Scrambler/Descrambler (100BASE-TX)	R/W	1 = Scrambler and descrambler disabled. 0 = Scrambler and descrambler enabled.	0
8	Bypass NRZI/MLT3 Encoder/Decoder (100BASE-TX)	R/W	1 = Bypass NRZI/MLT3 encoder and decoder. 0 = Normal operation.	0
7	Bypass Receive Symbol Alignment (100BASE-TX)	R/W	1 = 5B receive symbols not aligned. 0 = Receive symbols aligned to 5B boundaries.	0
6	Reset Scrambler (100BASE-TX)	R/W SC	1 = Reset scrambler to initial state. 0 = Normal scrambler operation.	0
5	Enable LED Traffic mode	R/W	1 = LED traffic mode enabled. 0 = LED traffic mode disabled.	0
4	Force LEDs On	R/W	1 = Force all LEDs into on-state except for LEDs programmed to INTR mode. 0 = Normal LED operation.	0
3	Force LEDs Off	R/W	1 = Force all LEDs into off-state except for LEDs programmed to INTR mode. 0 = Normal LED operation.	0
2	Reserved	R/W	Write as 0, ignore on read.	0
1	Reserved	R/W	Write as 0, ignore on read.	0
0	Reserved	R/W	Write as 0, ignore on read.	0

Disable Automatic MDI Crossover

The automatic MDI crossover function can be disabled by setting bit 14 = 1. When this bit is cleared, the NetXtreme II performs the automatic MDI crossover function.

Transmit Disable

The transmitter can be disabled by setting bit 13 = 1. The copper transmitter outputs are forced into a high impedance state.

Interrupt Disable

When this bit is set = 1, the INTR LED pin is forced to its inactive state.

Force Interrupt

When this bit is set = 1, the INTR LED pin is forced to its active state.

Bypass 4B/5B Encoder/Decoder (100BASE-TX)

The 100BASE-TX 4B/5B encoder/decoder can be bypassed by setting bit 10 = 1. The transmitter sends 5B codes from the TX_ER and TXD[3:0] pins directly to the scrambler. TX_EN is ignored and frame encapsulation (insertion of J/K and T/R codes) is not performed. The receiver places descrambled and aligned 5B codes onto the RX_ER and RXD[3:0] pins. CRS is still asserted when a valid frame is received.

Bypass Scrambler/Descrambler (100BASE-TX)

The 100BASE-TX stream cipher function can be disabled by setting bit 9 = 1. The stream cipher function can be re-enabled by writing a 0 to this bit.

Bypass MLT3 Encoder/Decoder (100BASE-TX)

The 100BASE-TX MLT3 encoder and decoder can be bypassed by setting bit 8 = 1. NRZ data is transmitted and received on the cable. The MLT3 encoder can be re-enabled by clearing this bit.

Bypass Receive Symbol Alignment (100BASE-TX)

100BASE-TX receive symbol alignment can be bypassed by setting bit 7 = 1. When used in conjunction with the bypass 4B/5B encoder/decoder bit, unaligned 5B codes are placed directly on the RX_ER and RXD[3:0] pins.

Reset Scrambler (100BASE-TX)

When bit 6 is set = 1, the NetXtreme II resets the scrambler to an all 1 state. This bit is self-clearing and always returns 0 when read.

Enable LED Traffic Mode

When bit 5 is set = 1, the NetXtreme II enables the LED Traffic mode for ACTIVITY, RCVLED and XMITLED. When this bit is cleared, the NetXtreme II disables the LED Traffic mode.

Force LEDs On

When bit 4 is set = 1, the NetXtreme II forces all LEDs into the ON state, except for LEDs programmed to INTR mode. When this bit is cleared, the NetXtreme II resets all LEDs to normal operation.

Force LEDs Off

When bit 3 is set = 1, the NetXtreme II forces all LEDs into the OFF state, except for LEDs programmed to INTR mode. When this bit is cleared, the NetXtreme II resets all LEDs to normal operation.

1000BASE-T/100BASE-TX/10BASE-T PHY EXTENDED STATUS REGISTER (ADDRESS 11H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0.

Table 563: 1000BASE-T/100BASE-TX/10BASE-T PHY Extended Status Register (Address 11h)

Bit	Name	R/W	Description	Default
15	Auto-negotiation Base Page Selector Field Mismatch	RO LH	1 = Link Partner Base Page Selector Field Mismatched Advertised Selector field since last read. 0 = No mismatch detected since last read.	0
14	Ethernet@Wirespeed Downgrade	RO	1 = Auto-negotiation advertised speed downgraded. 0 = No advertised speed downgrade.	0
13	MDI Crossover State	RO	1 = Crossover MDI mode. 0 = Normal MDI mode.	0
12	Interrupt Status	RO	1 = Unmasked interrupt currently active. 0 = Interrupt cleared.	0
11	Remote Receiver Status	RO LL	1 = Remote receiver OK. 0 = Remote receiver not OK since last read.	0
10	Local Receiver Status	RO LL	1 = Local receiver OK. 0 = Local receiver not OK since last read.	0
9	Locked	RO	1 = Descrambler locked. 0 = Descrambler unlocked.	0
8	Link Status	RO	1 = Link pass. 0 = Link fail.	0
7	CRC Error Detected	RO LH	1 = CRC error detected. 0 = No CRC error since last read.	0
6	Carrier Extension Error Detected	RO LH	1 = Carrier extension error detected since last read. 0 = No carrier extension error since last read.	0
5	Bad SSD Detected (False Carrier)	RO LH	1 = Bad SSD error detected since last read. 0 = No bad SSD error since last read.	0
4	Bad ESD Detected (Premature End)	RO LH	1 = Bad ESD error detected since last read. 0 = No bad ESD error since last read.	0
3	Receive Error Detected	RO LH	1 = Receive error detected since last read. 0 = No receive error since last read.	0
2	Transmit Error Detected	RO LH	1 = Transmit error code received since last read. 0 = No transmit error code received since last read.	0
1	Lock Error Detected	RO LH	1 = Lock error detected since last read. 0 = No lock error since last read.	0
0	MLT3 Code Error Detected	RO LH	1 = MLT3 code error detected since last read. 0 = No MLT3 code error since last read.	0

Auto-Negotiation Base Page Selector Field Mismatch

When this bit is set = 1, the auto-negotiation base page selector did not match the Advertised Selector field since the previous read. When this bit reads back a 0, there is no Mismatched Page Selector field and Advertised Selector field.

Ethernet@Wirespeed Downgrade

The NetXtreme II returns a 1 in bit 14 when an Ethernet@Wirespeed downgrade has occurred.

MDI Crossover State

When the NetXtreme II is automatically switching the transmit and receive pairs to communicate with a remote device, the NetXtreme II returns a 1 in bit 13. This bit returns a 0 when the NetXtreme II is in normal MDI mode.

Interrupt Status

The NetXtreme II returns a 1 in bit 12 when any unmasked interrupt is currently active; otherwise, it returns a 0.

Remote Receiver Status

When the remote receiver status is good, the NetXtreme II returns a 1 in bit 11. When the NetXtreme II detects that the remote receiver status is not good, this bit is latched at 0 and remains so until the bit is read and the remote receiver status is good.

Local Receiver Status

When the local receiver status is good, the NetXtreme II returns a 1 in bit 10. When the NetXtreme II detects that the local receiver status is not good, this bit is latched at 0 and remains so until the bit is read and the remote receiver status is good.

Locked

The NetXtreme II returns a 1 in bit 9 when the descrambler is locked to the incoming data stream; otherwise, it returns a 0.

Link Status

The NetXtreme II returns a 1 in bit 8 when the device has established a link; otherwise, it returns a 0.

CRC Error Detected

The NetXtreme II returns a 1 in bit 7 if a CRC error has been detected since the last time this register was read; otherwise, it returns a 0.

Carrier Extension Error Detected

The NetXtreme II returns a 1 in bit 6 if a carrier extension error has been detected since the last time this register was read; otherwise, it returns a 0.

Bad SSD Detected (False Carrier)

The NetXtreme II returns a 1 in bit 5 if a bad start-of-stream error has been detected since the last time this register was read; otherwise, it returns a 0.

Bad ESD Detected (Premature End)

The NetXtreme II returns a 1 in bit 4 if a bad end-of-stream error has been detected since the last time this register was read; otherwise, it returns a 0.

Receive Error Detected

The NetXtreme II returns a 1 in bit 3 if a packet was received with an invalid code since the last time this register was read; otherwise, it returns a 0.

Transmit Error Detected

The NetXtreme II returns a 1 in bit 2 if a packet was received with a transmit error code since the last time this register was read; otherwise, it returns a 0.

Lock Error Detected

The NetXtreme II returns a 1 in bit 1 if the descrambler has lost lock since the last time this register was read; otherwise, it returns a 0.

MLT3 Code Error Detected

The NetXtreme II returns a 1 in bit 0 if an MLT3 coding error has been detected in the receive data stream since the last time this register was read; otherwise, it returns a 0.

1000BASE-T/100BASE-TX/10BASE-T RECEIVE ERROR COUNTER REGISTER (ADDRESS 12H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0
- “Auxiliary Control Register (Address 1Ch, Shadow 1Bh)” bit 9 = 0.

Table 564: 1000BASE-T/100BASE-TX/10BASE-T Receive Error Counter Register (Address 12h)

Bit	Name	R/W	Description	Default
15:0	Receive Error Counter	R/W CR	Number of non-collision packets with receive errors since last read.	0000h

Copper Receive Error Counter

When bit 9 = 0 in register 1Ch, shadow 1Bh, this counter increments each time the NetXtreme II receives a 1000BASE-T, 100BASE-TX, or 10BASE-T non-collision packet containing at least one receive error. This counter freezes at the maximum value of FFFFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T FALSE CARRIER SENSE COUNTER REGISTER (ADDRESS 13H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0
- “Auxiliary Control Register (Address 1Ch, Shadow 1Bh)”, bit 9 = 0
- “Test Register 1 (Address 1Eh)”, bit 14 = 0.

**Table 565: 1000BASE-T/100BASE-TX/10BASE-T False Carrier Sense Counter Register
(Address 13h)**

Bit	Name	R/W	Description	Default
15:8	Reserved	RO	Ignore on read.	00h
7:0	False Carrier Sense Counter	R/W CR	Number of false carrier sense events since last read.	00h

Copper False Carrier Sense Counter

When bit 9 = 0 in register 1Ch, shadow 1Bh and bit 14 = 0 in register 1Eh, the False Carrier Sense Counter increments each time the NetXtreme II detects a 1000BASE-T, 100BASE-TX, or 10BASE-T false carrier sense on the receive input. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T PACKETS RECEIVED WITH TRANSMIT ERROR CODES COUNTER REGISTER (ADDRESS 13H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0
- “Auxiliary Control Register (Address 1Ch, Shadow 1Bh)”, bit 9 = 0
- “Test Register 1 (Address 1Eh)”, bit 14 = 1.

Table 566: 1000BASE-T/100BASE-TX/10BASE-T Transmit Error Code Counter Register (Address 13h)

Bit	Name	R/W	Description	Default
15:8	Reserved	RO	Write as 0, ignore on read.	00h
7:0	Transmit Error Code Counter	R/W CR	Number of packets received with transmit error codes since last read.	00h

Packets Received with Transmit Error Codes Counter

When bit 9 = 0 in register 1Ch, shadow 1Bh, and when bit 14 = 1 in register 1Eh, Packets Receive with Transmit Error Codes Counter increments each time the NetXtreme II detects a 1000BASE-T, 100BASE-TX, or 10BASE-T packet with a transmit error code violation. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T RECEIVER NOT_OK COUNTER REGISTER (ADDRESS 14H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0
- “Auxiliary Control Register (Address 1Ch, Shadow 1Bh)”, bit 9 = 0
- “Test Register 1 (Address 1Eh)”, bit 15 = 0.

**Table 567: 1000BASE-T/100BASE-TX/10BASE-T Receiver NOT_OK Counter Register
(Address 14h)**

Bit	Name	R/W	Description	Default
15:8	Local Receiver NOT_OK Counter	R/W CR	Number of times local receiver was NOT_OK since last read.	00h
7:0	Remote Receiver NOT_OK Counter	R/W CR	Number of times NetXtreme II detected that the remote receiver was NOT_OK since last read.	00h

Copper Local Receiver NOT_OK Counter

When bit 9 = 0 in register 1Ch, shadow 1Bh and bit 15 = 0 in register 1Eh, this counter increments each time the 1000BASE-T, 100BASE-TX, or 10BASE-T local receiver enters the NOT_OK state. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

Copper Remote Receiver NOT_OK Counter

When bit 9 = 0 in register 1Ch, shadow 1Bh and bit 15 = 0 in register 1Eh, this counter increments each time the 1000BASE-T, 100BASE-TX, or 10BASE-T remote receiver enters the NOT_OK state. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

1000BASE-T/100BASE-TX/10BASE-T RECEIVE CRC COUNTER REGISTER (ADDRESS 14H)

This register is enabled by writing to:

- “Mode Control Register (Address 1Ch, Shadow 1Fh)” bit 0 = 0
- “Auxiliary Control Register (Address 1Ch, Shadow 1Bh)”, bit 9 = 0
- “Test Register 1 (Address 1Eh)”, bit 15 = 1.

**Table 568: 1000BASE-T/100BASE-TX/10BASE-T CRC Counter Register
(Address 14h)^a**

Bit	Name	R/W	Description	Default
15:0	Receive CRC Counter	R/W CR	Number of times Receive CRC errors were detected.	00h

a. Bits[15:0] of this register become the 1000BASE-T, 100BASE-TX, or 10BASE-T Receive CRC Counter when register 1Ch, shadow 1Bh, bit 9 = 0 and register 1Eh bit 15 = 1.

Copper CRC Counter

When bit 9 = 0 in register 1Ch, shadow 1Bh and bit 15 = 1 in register 1Eh, this counter increments each time the 1000BASE-T, 100BASE-TX, or 10BASE-T detects a receive CRC error. This counter freezes at the maximum value of FFh. The counter automatically clears when read.

EXPANSION REGISTER ACCESS REGISTER (ADDRESS 17H)

Table 569: Expansion Register Access Register (Address 17h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Expansion Register Select	R/W	1111 = Expansion register selected.	0
10		R/W	0000 = Expansion register not selected.	0
9		R/W		0
8		R/W	All Others = Reserved (do not use).	0
7	Expansion Register Accessed	R/W	Sets the Expansion register number accessed when read/write to register 15h.	0
6		R/W		0
5		R/W		0
4		R/W		0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		0

Expansion Register Select

Setting bits[11:8] to 1111 enables the reading from and writing to the Expansion registers in conjunction with register 15h. These bits should be cleared after the Expansion registers are accessed or when the Expansion registers are not being accessed. See “[Expansion Registers](#)” on page 833 for Expansion register detail.

Expansion Register Accessed

Bits[7:0] determine which Expansion register is accessed. The Expansion registers can be accessed through register 15h when bits[11:8] of this register are set to 1111. The available expansion registers are listed in [Table 570](#).

Table 570: Expansion Register Select Values

Expansion Register	Register Name
00h	"Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)"
04h	"Expansion Register 04h: Multicolor LED Selector Register (Address 15h)"
05h	"Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)"
06h	"Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)"
10h	"Expansion Register 10h: Cable Diagnostic Controls and Basic Status Register (Address 15h)"
11h	"Expansion Register 11h: Cable Diagnostic Results Register (Address 15h)"
12h	"Expansion Register 12h: Cable Diagnostic Lengths Register (Address 15h)"
13h	"Expansion Register 13h: Cable Diagnostic Lengths Register (Address 15h)"
44h	"Expansion Register 44h: SGMII Lineside/Loopback Control Register (Address 15h)"
52h	"Expansion Register 52h: SGOUT \pm and SCLK \pm Control Register (Address 15h)"

AUXILIARY CONTROL SHADOW VALUE ACCESS REGISTER (ADDRESS 18H)

Available 18h registers are listed in [Table 571](#). [Table 572](#) and [Table 573](#) show the procedures for reading and writing register 18 shadow registers.

Table 571: Auxiliary Control Shadow Value Access Register (Address 18h)

Shadow Value	Register Name
000	"1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register (Address 18h, Shadow 0h)"
001	"10BASE-T Register (Address 18h, Shadow 1h)"
010	"1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register (Address 18h, Shadow 2h)"
100	"1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register (Address 18h, Shadow 4h)"
111	"1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register (Address 18h, Shadow 7h)"

Table 572: Reading Register 18h

Register Reads/Writes	Description
Write register 18h, bits[2:0] = 111	This selects the Miscellaneous Control register, shadow 7h. All reads must be done through the Miscellaneous Control register.
Bit 15 = 0	This allows only bits[14:12] and bits[2:0] to be written.
Bits[14:12] = zzz	This selects shadow register zzz to be read.
Bits[11: 3] = <don't care>	When bit 15 = 0, these bits are ignored.
Bits[2:0] = 111	This sets the Shadow Register Select to 111 (Miscellaneous Control register).
Read register 18h	Data read back is the value from shadow register zzz.

Table 573: Writing Register 18h

Register Writes	Description
Set Bits[15:3] = Preferred write values	Bits[15:3] contain the desired bits to be written to.
Set Bits[2:0] = yyy	This enables shadow register yyy to be written. For shadow 7h, bit 15 must also be written.

**1000BASE-T/100BASE-TX/10BASE-T AUXILIARY CONTROL REGISTER (ADDRESS 18H,
SHADOW 0H)**

**Table 574: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Control Register
(Address 18h, Shadow 0h)**

Bit	Name	R/W	Description	Default
15	External Loopback	R/W	1 = External loopback enabled. 0 = Normal operation.	0
14	Receive Extended Packet Length	R/W	1 = Allow reception of extended length packets. 0 = Allow only reception of normal length Ethernet packets.	0
13	Reserved	R/W	Write as 0, ignore when read.	0
12	Reserved	R/W	Write as 0, ignore when read.	0
11	Enable State Machine DSP Clock	R/W	1 = Enable DSP clock 0 = Normal operation	0
10	Reserved	R/W	Write as 1, ignore on read.	1
9	Reserved	R/W	Write as 0, ignore on read.	0
8	Reserved	R/W	Write as 0, ignore on read.	0
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Reserved	R/W	Write as 0, ignore on read.	0
5	Reserved	R/W	Ignore when read.	0
4	Reserved	R/W	Ignore when read.	0
3	Reserved	R/W	Write as 0, ignore on read.	0
2	Shadow Register Select	R/W	0h = Auxiliary Control register.	0
1		R/W	1h = 10BASE-T register.	0
0		R/W	2h = Power/MII Control register. 4h = Miscellaneous Test register. 7h = Miscellaneous Control register.	0

External Loopback

When bit 15 = 1, external loopback operation is enabled. When this bit is cleared, normal operation resumes.

Receive Extended Packet Length

When bit 14 = 1, the NetXtreme II can receive packets up to 10 KB in length when in SGMII mode. When this bit is cleared, the NetXtreme II only receives packets up to 4.5 KB in length.

Shadow Register Select

See [Table 572 on page 770](#) and [Table 573 on page 770](#) for information on reading from and writing to register 18h.

The register set shown above is for normal operation, obtained when the lower 3 bits are 000.

10BASE-T REGISTER (ADDRESS 18H, SHADOW 1H)

**Table 575: 10BASE-T Register
(Address 18h, Shadow 1h)**

Bit	Name	R/W	Description	Default
15	Manchester Code Error	RO LH	1 = Manchester code error (10BASE-T). 0 = No Manchester code error.	0
14	EOF Error	RO LH	1 = EOF error detected (10BASE-T). 0 = No EOF error detected.	0
13	Polarity Error	RO	1 = Channel polarity inverted. 0 = Channel polarity correct.	0
12	Block RX_DV Extension (IPG)	R/W	1 = Block RX_DV for four additional RXC cycles for IPG. 0 = Normal operation.	0
11	10BASE-T TXC Invert Mode	R/W	1 = Invert TXC output. 0 = Normal operation.	0
10	Reserved	R/W	Write as 0, ignore on read.	0
9	Jabber Disable	R/W	1 = Jabber function disabled. 0 = Jabber function enabled.	0
8	Reserved	R/W	Write as 1, ignore on read.	1
7	Reserved	R/W	Write as 0, ignore on read.	0
6	10BASE-T Echo Mode	R/W	1 = Echo transmit data to receive data. 0 = Normal operation.	0
5	SQE Enable Mode	R/W	1 = Enable SQE. 0 = Disable SQE.	0
4	10BASE-T No Dribble	R/W	1 = Correct 10BASE-T dribble nibble. 0 = Normal operation.	0
3	Reserved	R/W	Write as 0, ignore on read.	0
2	Shadow Register Select	R/W	0h = Auxiliary Control register. 1h = 10BASE-T register.	0
1		R/W	2h = Power/MII Control register.	0
0		R/W	4h = Miscellaneous Test register. 7h = Miscellaneous Control register.	1

Manchester Code Error

When a Manchester code violation is received, bit 15 returns a 1. This bit is valid only during 10BASE-T operation.

EOF Error

When the end-of-frame (EOF) sequence was improperly received (or not received at all), bit 14 returns a 1. This bit is valid only during 10BASE-T operation.

Polarity Error

When an analog input polarity error has been detected and corrected, bit 13 returns a 1. This bit is valid only during 10BASE-T operation.

Block RX_DV Extension (IPG)

When bit 12 of the 10BASE-T register is set, blocking of the RX_DV signal is extended for an additional four RXC cycles to extend the IPG.

10BASE-T TXC Invert Mode

When bit 11 of the 10BASE-T register is set, the polarity of the 10BASE-T transmit clock is inverted. Clearing this bit restores normal transmit clock polarity. This bit is valid only during 10BASE-T operation.

Jabber Disable

Setting bit 9 allows the user to disable the jabber detect function defined in the IEEE standard. When a transmission request has exceeded a maximum time limit, this function shuts off the transmitter. Clearing this bit or resetting the chip restores normal operation. Reading this bit returns the value of jabber detect disable. This bit is valid only during 10BASE-T operation.

10BASE-T Echo Mode

When bit 6 is enabled during 10BASE-T half-duplex transmit operation, the transmitted data is replicated on the receive data pins and the TXEN signal echoes on the RX_DV pin. The TXEN signal also echoes on the CRS pin and CRS deassertion directly follows the TXEN deassertion.

SQE Enable Mode

Setting bit 5 of the 10BASE-T register enables SQE mode. Clearing disables it. This bit is valid only during 10BASE-T operation.

10BASE-T No Dribble

When bit 4 is set, the PHY rounds down to the nearest nibble when dribble bits are present on the 10BASE-T input stream.

Shadow Register Select

See [Table 572](#) and [Table 573 on page 770](#) for information on reading from and writing to register 18h.

1000BASE-T/100BASE-TX/10BASE-T POWER/MII CONTROL REGISTER (ADDRESS 18H, SHADOW 2H)

*Table 576: 1000BASE-T/100BASE-TX/10BASE-T Power/MII Control Register
(Address 18h, Shadow 2h)*

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Reserved	R/W	Write as 0, ignore on read.	0
10:7	2.5V Voltage Regulator Output Voltage	R/W	These 4 bits are used to adjust the output voltage from approximately -16% to +14% in 2% steps. 0000 +14% 0100 +12% 1000 +10% 1100 +8% 0001 +6% 0101 +4% 1001 +2% 1101 0% (Default) 0010 -2% 0110 -4% 1010 -6% 1110 -8% 0011 -10% 0111 -12% 1011 -14% 1111 -16%	1101
6	Reserved	R/W	Write as 1, ignore on read.	1
5	Super Isolate (Copper Only)	R/W	1 = Isolate mode with no link pulses transmitted. 0 = Normal operation.	1
4	Reserved	R/W	Write as 0, ignore on read.	0
3	Reserved	R/W	Write as 0, ignore on read.	0
2	Shadow Register Select	R/W	0h = Auxiliary Control register.	0
1		R/W	1h = 10BASE-T register.	1
0		R/W	2h = Power/MII Control register. 4h = Miscellaneous Test register. 7h = Miscellaneous Control register.	0

Super Isolate (Copper Only)

Setting bit 5 = 1, places the NetXtreme II into the Super Isolate mode.

When in SGMII mode, the transmit and receive functions on the Copper Media Dependent Interface are disabled, the SGMII interface is in auto-negotiation mode, and link is established with the switch if the switch sends back an acknowledgement to the PHY. If the switch does not send back an acknowledgement, the PHY remains in auto-negotiation mode. Any data received from the switch is ignored, and no data is sent from the PHY to the switch. Only auto-negotiation link code words pass between the PHY and the switch because the PHY's Copper MDI is disabled, and the PHY communicates to the switch that there is no Copper link.

Shadow Register Select

See [Table 572](#) and [Table 573 on page 770](#) for information on reading from and writing to register 18h.

1000BASE-T/100BASE-TX/10BASE-T MISCELLANEOUS TEST REGISTER (ADDRESS 18H, SHADOW 4H)

**Table 577: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Test Register
(Address 18h, Shadow 4h)**

Bit	Name	R/W	Description	Default
15	Lineside [Remote] Loopback Enable	R/W	1 = Enable Lineside [Remote] Loopback. 0 = Disable loopback.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Reserved	R/W	Write as 0, ignore on read.	0
10	10BASE-T Wakeup	R/W	1 = Enable 10BASE-T Wakeup. 0 = Disable loopback.	0
9	Reserved	R/W	Write as 0, ignore on read.	0
8	Reserved	R/W	Write as 0, ignore on read.	0
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Reserved	R/W	Write as 0, ignore on read.	0
5	Reserved	R/W	Write as 0, ignore on read.	0
4	Swap RX MDIX	R/W	1 = RX and TX operate on same pair. 0 = Normal operation.	0
3	10BASE-T Halfout	R/W	1 = Transmit 10BASE-T at half amplitude. 0 = Normal operation.	0
2	Shadow Register Select	R/W	0h = Auxiliary Control register. 1h = 10BASE-T register.	1
1		R/W	2h = Power/MII Control register.	0
0		R/W	4h = Miscellaneous Test register. 7h = Miscellaneous Control register.	0

Lineside [Remote] Loopback Enable

Setting bit 15 = 1 enables Lineside [Remote] Loopback of the copper receive packet back out through the MDI transmit path.

10BASE-T Wakeup

Setting bit 10 = 1 enables the 10BASE-T link pulse transmitter to function properly in capacitive coupling applications.

Swap RX MDIX

When bit 4 = 1, the transmitter and receiver operate on the same twisted pair. This function is for use in a test mode where the transmitter output is detected by the receiver attached to the same pair.

10BASE-T Halfout

When operating in 10BASE-T mode, setting bit 3 = 1 reduces the output of the transmitter to half of its normal amplitude. Clearing this bit restores full amplitude operation. This function is used in a test mode where an un-terminated output generates a signal with twice the amplitude of a terminated output.

Shadow Register Select

See [Table 572](#) and [Table 573 on page 770](#) for information on reading from and writing to register 18h.

1000BASE-T/100BASE-TX/10BASE-T MISCELLANEOUS CONTROL REGISTER (ADDRESS 18H, SHADOW 7H)

**Table 578: 1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register
(Address 18h, Shadow 7h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W SC	1 = Write bits[14:3]. 0 = Only write bits[14:12].	0
14	Shadow Register Read Selector	R/W	These bits are written when bit 15 is not set. This sets the shadow for address 18h register read.	0
13		R/W	000 = Normal operation	0
12		R/W	001 = 10BASE-T register 010 = Power Control register 100 = Miscellaneous Test register 111 = Miscellaneous Control register	0
11	Packet Counter Mode	R/W	1 = Receive packet counter. 0 = Transmit packet counter.	0
10	Reserved	R/W	Write as 0, ignore on read.	0
9	Force Auto-MDIX Mode	R/W	1 = Auto-MDIX is enabled when auto-negotiation is disabled. 0 = Auto-MDIX is disabled when auto-negotiation is disabled.	0
8	Reserved	R/W	Write as 0, ignore on read.	0
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Reserved	R/W	Write as 1, ignore on read.	1
5	Reserved	R/W	Write as 1, ignore on read.	1
4	Ethernet@Wirespeed Enable	R/W	1 = Enable Ethernet@Wirespeed. 0 = Disable Ethernet@Wirespeed.	1
3	MDIO All PHY Select	R/W	1 = The PHY ports accepts MDIO writes to PHY Address = 00000. 0 = Normal operation.	0
2	Shadow Register Select	R/W	0h = Auxiliary Control register.	1
1		R/W	1h = 10BASE-T register.	1
0		R/W	2h = Power/MII Control register. 4h = Miscellaneous Test register. 7h = Miscellaneous Control register.	1

Write Enable

When bit 15 is set = 1 then bits[14:3] of this register can be modified. Bits[2:0] and [14:12] can always be written, regardless of the state of bit 15.

Shadow Register Read Selector

Bits[14:12] are written, regardless of the value of bit 15. These bits determine the shadow for an MII register 18h read operation.

See [Table 572](#) and [Table 573 on page 770](#) for information on reading from and writing to register 18h.

Packet Counter Mode

Bit 11 sets the packet counter mode in Expansion register 00h. If bit 11 = 1, it counts the number of receive packets. If bit 11 = 0, it counts the number of transmit packets.

Force Auto-MDIX Mode

Setting bit 9 = 1 enables Auto-MDIX mode while auto-negotiation is disabled. The default is to disable the Auto-MDIX function when auto-negotiation is disabled.

Ethernet@Wirespeed Enable

When bit 4 = 1, Ethernet@Wirespeed mode is enabled. If the link cannot be established within two to nine attempts (the number of attempts is set by bits[4:2] in register 1Ch, shadow 04h), the NetXtreme II downgrades its advertised abilities and again tries to establish a link. When bit 4 is cleared, the NetXtreme II advertises its abilities according to registers 04h and 09h.

MDIO All PHY Select

When bit 3 = 1, a MDIO write operation with PHY Address = 00000 is accepted in addition to the real address of the PHY. By setting this bit to a 1 for all eight ports, a single write to PHY address 00000 writes all eight ports.

Shadow Register Select

See [Table 572](#) and [Table 573 on page 770](#) for information on reading from and writing to register 18h.

1000BASE-T/100BASE-TX/10BASE-T AUXILIARY STATUS SUMMARY REGISTER (ADDRESS 19H)

Table 579: 1000BASE-T/100BASE-TX/10BASE-T Auxiliary Status Summary Register (Address 19h)

Bit	Name	R/W	Description	Default
15	Auto-negotiation Complete	RO	1 = Auto-negotiation complete. 0 = Auto-negotiation in progress.	0
14	Auto-negotiation Complete Acknowledge	RO LH	1 = Entered Auto-negotiation Link OK Check state. 0 = State not entered since last read.	0
13	Auto-negotiation Acknowledge Detect	RO LH	1 = Entered Auto-negotiation Acknowledge Detect state. 0 = State not entered since last read.	0
12	Auto-negotiation Ability Detect	RO LH	1 = Entered auto-negotiation ability detect state. 0 = State not entered since last read.	0
11	Auto-negotiation Next Page Wait	RO LH	1 = Entered auto-negotiation Next Page wait state. 0 = State not entered since last read.	0
10	Auto-negotiation HCD	RO	111 = 1000BASE-T full-duplex ^a	0
9	Current Operating Speed and Duplex Mode	RO	110 = 1000BASE-T half-duplex ^a	0
8		RO	101 = 100BASE-TX full-duplex ^a 100 = 100BASE-T4 011 = 100BASE-TX half-duplex ^a 010 = 10BASE-T full-duplex ^a 001 = 10BASE-T half-duplex ^a 000 = No highest common denominator or auto-negotiation not complete.	0
7	Parallel Detection Fault	RO LH	1 = Parallel link fault detected. 0 = Parallel link fault not detected.	0
6	Remote Fault	RO	1 = Link partner has detected remote fault. 0 = Link partner has not detected remote fault.	0
5	Auto-negotiation Page Received	RO LH	1 = New page has been received from link partner. 0 = New page has not been received.	0
4	Link Partner Auto-negotiation Ability	RO	1 = Link partner has auto-negotiation capability. 0 = Link partner does not perform auto-negotiation.	0
3	Link Partner Next Page Ability	RO	1 = Link partner has Next Page capability. 0 = Link partner does not have Next Page capability.	0
2	Link Status	RO	1 = Link is up (link pass state). 0 = Link is down (link fail state).	0
1	Pause Resolution—Receive Direction	RO	1 = Enable pause receive. 0 = Disable pause receive.	0
0	Pause Resolution—Transmit Direction	RO	1 = Enable pause transmit. 0 = Disable pause transmit.	0

a. Indicates the negotiated HCD when Auto-negotiation Enable = 1, or indicates the manually selected speed and duplex mode when Auto-negotiation Enable = 0

Auto-Negotiation Complete

When auto-negotiation is complete, the NetXtreme II returns a 1 in bit 15. This bit returns a 0 while auto-negotiation is in progress.

Auto-Negotiation Complete Acknowledge

The NetXtreme II returns a 1 in bit 14 when the auto-negotiation state machine has entered the Link OK Check state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation Acknowledge Detect

The NetXtreme II returns a 1 in bit 13 when the auto-negotiation state machine has entered the Acknowledge Detect state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation Ability Detect

The NetXtreme II returns a 1 in bit 12 when the auto-negotiation state machine has entered the Ability Detect state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation Next Page Wait

The NetXtreme II returns a 1 in bit 11 when the auto-negotiation state machine has entered the Next Page Wait state since the last time this register was read; otherwise, it returns a 0.

Auto-Negotiation HCD (Current Operating Speed and Duplex Mode)

Bits[10:8] report the mode of operation negotiated between the NetXtreme II and its link partner. As reported by bit 15, these bits return 000 until auto-negotiation is completed. When the auto-negotiation function has been disabled, bits[10:8] report the manually selected mode of operation when register 18h, shadow 7h, bit 9 = 0. If auto-negotiation is disabled and register 18h, shadow 7h, bit 9 = 1, then bits[10:8] = 000.

Parallel Detection Fault

When a parallel detection fault has occurred in the auto-negotiation state machine, bit 7 returns a 1. When a parallel detection fault occurs, this bit is latched to a 1 and remains so until the next register read. This bit returns a 0 when a parallel detection fault has not occurred since the last time it was read.

Remote Fault

The NetXtreme II returns a 1 in bit 6 when the link partner has detected a remote fault; otherwise, it returns a 0.

Auto-Negotiation Page Received

The NetXtreme II returns a 1 in bit 5 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

Link Partner Auto-Negotiation Ability

The NetXtreme II returns a 1 in bit 4 of the when the link partner is known to have auto-negotiation capability. Before any auto-negotiation information is exchanged, or if the link partner does not comply with IEEE auto-negotiation, the bit returns a 0.

Link Partner Next Page Ability

The NetXtreme II returns a 1 in bit 3 when the link partner needs to transmit Next Page information; otherwise, it returns a 0.

Link Status

The NetXtreme II returns a 1 in bit 2 when the link status is good; otherwise, it returns a 0.

Pause Resolution—Receive Direction and Transmit Direction

When auto-negotiation has completed, the NetXtreme II returns the result of the pause resolution function for full-duplex flow control on bits[1:0]. When bit 1 returns a 1, the link partner can send pause frames toward the local device. When bit 0 returns a 1, pause frames can be transmitted by the local device to the link partner. These bits are only guaranteed to be valid when bit 15 of the Auxiliary Status Summary register is 1.

1000BASE-T/100BASE-TX/10BASE-T INTERRUPT STATUS REGISTER (ADDRESS 1Ah)**Table 580: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Status Register (Address 1Ah)**

Bit	Name	R/W	Description	Default
15	Signal Detect/Energy Detect Change	RO LH	1 = Filtered fiber signal detect change or energy detect change since last read (enabled by register 1Ch, shadow 05h, bit 5 = 1). 0 = Interrupt cleared.	0
14	Illegal Pair Swap	RO LH	1 = Illegal pair swap detected. 0 = Interrupt cleared.	0
13	MDIX Status Change	RO LH	1 = MDIX status changed since last read. 0 = Interrupt cleared.	0
12	Exceeded High Counter Threshold	RO	1 = Value in one or more counters is above 32K. 0 = All counters below 32K.	0
11	Exceeded Low Counter Threshold	RO	1 = Value in one or more counters is above 128K. 0 = All counters below 128K.	0
10	Auto-negotiation Page Received	RO LH	1 = Page received since last read. 0 = Interrupt cleared.	0
9	No HCD Link	RO LH	1 = Negotiated HCD, did not establish link. 0 = Interrupt cleared.	0
8	No HCD	RO LH	1 = Auto-negotiation returned HCD = none. 0 = Interrupt cleared.	0
7	Negotiated Unsupported HCD	RO LH	1 = Auto-negotiation HCD not supported by the NetXtreme II. 0 = Interrupt cleared.	0
6	Scrambler Synchronization Error	RO LH	1 = Scrambler synchronization error occurred since last read. 0 = Interrupt cleared.	0
5	Remote Receiver Status Change	RO LH	1 = Remote receiver status changed since last read. 0 = Interrupt cleared.	0
4	Local Receiver Status Change	RO LH	1 = Local receiver status changed since last read. 0 = Interrupt cleared.	0
3	Duplex Mode Change	RO LH	1 = Duplex mode changed since last read. 0 = Interrupt cleared.	0
2	Link Speed Change	RO LH	1 = Link speed changed since last read. 0 = Interrupt cleared.	0
1	Link Status Change	RO LH	1 = Link status changed since last read. 0 = Interrupt cleared.	0
0	Receive CRC Error	RO LH	1 = Receive CRC error occurred since last read. 0 = Interrupt cleared.	0

The INTR LED output is asserted when any bit in 1000BASE-T/100BASE-TX/10BASE-T Interrupt Status register is set, and the corresponding bit in the 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask register is cleared.

Signal Detect/Energy Detect Change

This bit indicates the fiber signal detect (bit 4 in register 1Ch, shadow 1Fh) or the copper energy detect (bit 5 in register 1Ch, shadow 1Fh) has changed since the last read. Enabled by register 1Ch, shadow 05h, bit 5 = 1.

Illegal Pair Swap

The NetXtreme II returns a 1 in bit 14 when an uncorrectable pair swap error on the twisted-pair cable has been detected since the last time this register was read; otherwise, it returns a 0.

MDIX Status Change

The NetXtreme II returns a 1 in bit 13 when a link pulse or 100BASE-TX carrier was detected on a different pair than previously detected since the last time this register was read; otherwise, it returns a 0.

Exceeded High Counter Threshold

The NetXtreme II returns a 1 in bit 12 when one or more of the counters in registers 12h–14h is above 32 000; otherwise, it returns a 0.

Exceeded Low Counter Threshold

The NetXtreme II returns a 1 in bit 11 when one or more of the counters in registers 12h–14h is above 128 000; otherwise, it returns a 0.

Auto-Negotiation Page Received

The NetXtreme II returns a 1 in bit 10 when a new link code word has been received from the link partner since the last time this register was read; otherwise, it returns a 0.

No HCD Link

When the negotiated HCD is not able to establish a link, bit 9 returns a 1 by the NetXtreme II. The bit is cleared when the register is read.

No HCD

When auto-negotiation returns No HCD, bit 8 returns a 1 by the NetXtreme II. The bit is cleared when the register is read.

Negotiated Unsupported HCD

When the auto-negotiation HCD is not supported by the NetXtreme II, bit 7 returns a 1. The NetXtreme II does not support 100BASE-T4. The bit is cleared when the register is read.

Scrambler Synchronization Error

The NetXtreme II returns a 1 in bit 6 when a scrambler synchronization error has been detected since the last time this register was read; otherwise, it returns a 0.

Remote Receiver Status Change

The NetXtreme II returns a 1 in bit 5 when the remote receiver status has changed since the last time this register was read; otherwise, it returns a 0.

Local Receiver Status Change

The NetXtreme II returns a 1 in bit 4 when the local receiver status has changed since the last time this register was read; otherwise, it returns a 0.

Duplex Mode Change

The NetXtreme II returns a 1 in bit 3 when the duplex mode has changed since the last time this register was read; otherwise, it returns a 0.

Link Speed Change

The NetXtreme II returns a 1 in bit 2 when the link speed has changed since the last time this register was read; otherwise, it returns a 0.

Link Status Change

The NetXtreme II returns a 1 in bit 1 when the link status has changed since the last time this register was read; otherwise, it returns a 0.

Receive CRC Error

The NetXtreme II returns a 1 in bit 0 when a receive CRC error has been detected since the last time this register was read; otherwise, it returns a 0.

1000BASE-T/100BASE-TX/10BASE-T INTERRUPT MASK REGISTER (ADDRESS 1BH)**Table 581: 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask Register (Address 1Bh)**

Bit	Name	R/W	Description	Default
15	Signal Detect/Energy Detect Change (enabled by register 1Ch, shadow 05h, bit 5 = 1)	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
14	Illegal Pair Swap	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
13	MDIX Status Change	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
12	Exceeded High Counter Threshold	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
11	Exceeded Low Counter Threshold	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
10	Auto-negotiation Page Received	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
9	HCD No Link	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
8	No HCD	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
7	Negotiated Unsupported HCD	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
6	Scrambler Synchronization Error	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
5	Remote Receiver Status Change	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
4	Local Receiver Status Change	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
3	Duplex Mode Change	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
2	Link Speed Change	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
1	Link Status Change	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1
0	CRC Error	R/W	1 = Interrupt masked, status bits operate normally. 0 = Interrupt enabled, status bits operate normally.	1

Interrupt Mask Vector

When bit *n* of the 1000BASE-T/100BASE-TX/10BASE-T Interrupt Mask register is written to 1, the interrupt corresponding to the same bit in the Interrupt Status register is masked. The status bits still operate normally when the interrupt is masked, but do not generate an interrupt output. When this bit is written to 0, the interrupt is unmasked.

REGISTER 1Ch ACCESS

Reading from and writing to register 1Ch is through register 1Ch bits[15:10]. Bits[14:10] set the shadow of register 1Ch, and bit 15 enables the writing of bits[9:0]. Setting bit 15 allows writing to bits[9:0] of register 1Ch. To read register 1C shadow zzzzz, first set writes to register 1Ch with bit 15 = 0, and bits[14:10] to zzzzz. The subsequent register read from register 1Ch contains the shadow zzzzz register value. [Table 582](#) lists all the register 1Ch shadows.

Table 582: Register 1Ch Shadows

Shadow Value	Register Name
00010	"1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register (Address 1Ch, Shadow 02h)"
00100	"1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register (Address 1Ch, Shadow 04h)"
00101	"1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register (Address 1Ch, Shadow 05h)"
01000	"1000BASE-T/100BASE-TX/10BASE-T LED Status Register (Address 1Ch, Shadow 08h)"
01001	"1000BASE-T/100BASE-TX/10BASE-T LED Control Register (Address 1Ch, Shadow 09h)"
01010	"Auto Power-Down Register (Address 1Ch, Shadow 0Ah)"
01011	"SLED_1 Register (Address 1Ch, Shadow 0Bh)"
01100	"SLED_2 Register (Address 1Ch, Shadow 0Ch)"
01101	"1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register (Address 1Ch, Shadow 0Dh)"
01110	"1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register (Address 1Ch, Shadow 0Eh)"
01111	"LED GPIO Control/Status Register (Address 1Ch, Shadow Value 0Fh)"
10101	"SGMII Status Register (Address 1Ch, Shadow 15h)"
10110	"Misc. 1000BASE-X Control 2 Register (Address 1Ch, Shadow 16h)"
11000	"1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h)"
11011	"Auxiliary Control Register (Address 1Ch, Shadow 1Bh)"
11111	"Mode Control Register (Address 1Ch, Shadow 1Fh)"

1000BASE-T/100BASE-TX/10BASE-T SPARE CONTROL 1 REGISTER (ADDRESS 1Ch, SHADOW 02h)

**Table 583: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register
(Address 1Ch, Shadow 02h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	00010 = Spare Control 1 register	0
13		R/W		0
12		R/W		0
11		R/W		1
10		R/W		0
9	100FX SD Status	R/W	1 = 100FX SD input active. 0 = 100FX SD input not-active.	0
8	Force 100FX SD On	R/W	1 = Force 100FX SD input active. 0 = Normal 100FX SD operation.	0
7	100FX SD Invert	R/W	1 = Fiber Signal Detect is an active low signal from optical module (RX_LOS). 0 = Fiber Signal Detect is an active high signal from optical module (SD). This is only valid for 100BASE-FX mode.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	Use Filtered Signal Detect	R/W	1 = Enable filter on Signal Detect input. 0 = Normal operation.	0
4	100FX Mode	R/W	1 = Enable 100FX mode on Copper pins. 0 = Normal operation.	0
3	Reserved	R/W	Write as 0, ignore when read.	0
2	Link Speed LED mode	R/W	1 = Enable Link Speed LED mode. LINKSPD[2:1] = speed 10: 1000BASE-T link 01: 100BASE-TX link 11: 10BASE-T or auto-negotiating 0 = Normal link LED mode.	0
1	Reserved	R/W	Write as 0, ignore when read.	1
0	Link LED Mode	R/W	1 = Enable link LED mode. LINKSPD[2:1] = speed 00: 1000BASE-T link 01: 100BASE-TX link 10: 10BASE-T link or no link SLAVE = Active low 10/100/1000BASE-T link 0 = Normal link LED mode.	0

Write Enable

Bit 15 = 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform a MDIO write with bit 15 = 0 and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 00010 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 register address 1Ch.

100FX SD Status

This bit is used to monitor the status of the SD input when in 100FX mode. When bit 9 = 1, the 100FX SD input is active.

Force 100FX SD Input ON

This bit is used to force the 100FX SD bit active. This feature is useful for applications where you do not want to connect a SD input to the PHY. When bit 8 = 1, the 100FX SD input is forced to the active state.

100FX SD Invert

Setting bit 7 = 1 allows the PHY to receive RX_LOS signals from the optical module. Setting bit 7 = 0 allows the PHY to receive SD signals from the optical module.

100BASE-FX Mode

Setting bit 4 = 1 enables 100BASE-FX mode. In addition to bit 4 being set, the PHY also needs to be forced to 100BASE-TX full- or half-duplex mode and the rise/fall times of the transmitter set to 0 ns.

Link Speed LED

Setting bit 2 = 1 enables the Link Speed Led mode.

Link LED Mode

Setting bit 0 = 1, enables Link LED mode. The LINKSPD2/LINKSPD1 are Link-Speed LED and SLAVE LED is LINK LED to indicate a link for 10BASE-T, 100BASE-TX or 1000BASE-T. When this bit is cleared, the LINKSPD2, LINKSPD1, and SLAVE operate in their normal modes.

**Table 584: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register
(Address 1Ch, Shadow 02h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	00010 = Spare Control 1 register	0
13		R/W		0
12		R/W		0
11		R/W		1
10		R/W		0



**Table 584: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 1 Register
(Address 1Ch, Shadow 02h) (Cont.)**

Bit	Name	R/W	Description	Default
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	Reserved	R/W	Write as 0, ignore when read.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	Reserved	R/W	Write as 0, ignore when read.	0
4	Reserved	R/W	Write as 0, ignore when read.	0
3	Reserved	R/W	Write as 0, ignore when read.	0
2	Link Speed LED mode	R/W	1 = Enable Link Speed LED mode. • <u>LINKSPD[2:1]</u> = speed • 10: 1000BASE-T link • 01: 100BASE-TX link • 11: 10BASE-T or auto-negotiating 0 = Normal link LED mode.	0
1	Reserved	R/W	Write as 0, ignore when read.	1
0	Link LED Mode	R/W	1 = Enable link LED mode. • <u>LINKSPD[2:1]</u> = speed • 00: 1000BASE-T link • 01: 100BASE-TX link • 10: 10BASE-T link or no link • <u>SLAVE</u> = Active low 10/100/1000BASE-T link 0 = Normal link LED mode.	0

**1000BASE-T/100BASE-TX/10BASE-T SPARE CONTROL 2 REGISTER (ADDRESS 1Ch,
SHADOW 04H)**

*Table 585: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 Register
(Address 1Ch, Shadow 04h)*

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0]. 0 = Read bits[9:0].	0
14	Shadow Register Selector	R/W	00100 = Spare Control 2 register.	0
13		R/W		0
12		R/W		1
11		R/W		0
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	Reserved	R/W	Write as 0, ignore when read.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	Reserved	R/W	Write as 0, ignore when read.	0
4	Ethernet@Wirespeed Retry Limit	R/W	000 = Downgrade after 2 failed auto-negotiation attempts. 001 = Downgrade after 3 failed auto-negotiation attempts. 010 = Downgrade after 4 failed auto-negotiation attempts. 011 = Downgrade after 5 failed auto-negotiation attempts. 100 = Downgrade after 6 failed auto-negotiation attempts. 101 = Downgrade after 7 failed auto-negotiation attempts. 110 = Downgrade after 8 failed auto-negotiation attempts. 111 = Downgrade after 9 failed auto-negotiation attempts.	0 1 1 1 1 1 1 1 1 1
1	Energy Detect on INTR LED Pin	R/W	1 = The INTR pin is driven high when the NetXtreme II detects energy on the twisted pair interface. Use LED selectors (register 1Ch, shadow 0Dh and 0Eh) and program to INTR mode. 0 = INTR LED pin performs the normal interrupt function.	0
0	Reserved	R/W	Write as 0, ignore when read.	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] must be set to 00100 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Spare Control 2 register.

Ethernet@Wirespeed Retry Limit

Bits[4:2] set the numbers of auto-negotiation attempts to link up prior to speed downgrade. The Ethernet@Wirespeed mode must be enabled for these bits to work.

Energy Detect on INTR LED Pin

The INTR pin is driven high when the NetXtreme II detects energy on the twisted pair interface. Set the LED selector register to enable INTR LED mode (1Ch, shadow 0Dh and/or 0Eh set bits[7:4] and or bits[3:0] to 0110, depending upon the LED being used). When Energy is detected (bit 5 = 1 in register 1Ch, shadow 1Fh) the INTR pin is driven high.

**1000BASE-T/100BASE-TX/10BASE-T SPARE CONTROL 3 REGISTER (ADDRESS 1Ch,
SHADOW 05H)**

**Table 586: 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 Register
(Address 1Ch, Shadow 05h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0]. 0 = Read bits[9:0].	0
14	Shadow Register Selector	R/W	00101 = Spare Control 3 register.	0
13		R/W		0
12		R/W		1
11		R/W		0
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	Reserved	R/W	Write as 0, ignore when read.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	SD/Energy Detect Change	R/W	1 = Interrupt based on energy detection or signal detect change. 0 = Normal operation.	0
4	Reserved	R/W	Write as 1, ignore when read.	1
3	Reserved	R/W	Write as 1, ignore when read.	1
2	Reserved	R/W	Write as 1, ignore when read.	1
1	DLL Auto Power-Down	R/W	1 = Auto power down of DLL is disabled. 0 = Auto power down of DLL is enabled.	1
0	CLK125 Output	R/W	1 = Enable CLK125 output. 0 = Disable CLK125 output.	1

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow register in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] must be set to 00101 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Spare Control 3 register.

SD/Energy Detect Change

Setting this bit = 1, asserts the INTR output pin when there is a change in status in bits[5:4] in register 1Ch, shadow 1Fh and bit 15 = 0 in register 1Bh.

DLL Auto Power-Down

Setting this bit = 0, enables the Auto Power-Down mode of the internal DLL. This feature enables additional power savings. This feature should only be used during Auto Power-Down mode.

CLK125 Output

Setting this bit = 1, enables CLK125 Output; clearing this bit disables CLK125 Output.

1000BASE-T/100BASE-TX/10BASE-T LED STATUS REGISTER (ADDRESS 1Ch, SHADOW 08H)

**Table 587: 1000BASE-T/100BASE-TX/10BASE-T LED Status Register
(Address 1Ch, Shadow 08h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01000 = LED Status register	0
13		R/W		1
12		R/W		0
11		R/W		0
10		R/W		0
9	Reserved	RO	Ignore on read.	0
8	<u>SLAVE</u> Indicator	RO	1 = Master mode. 0 = Slave mode.	0
7	<u>FDX</u> Indicator	RO	1 = Half-duplex mode. 0 = Full-duplex mode.	0
6	<u>INTR</u> Indicator	RO	1 = No active interrupt. 0 = Interrupt activated.	0
5	Reserved	RO	Ignore on read.	0
4	<u>LINKSPD</u> Indicator	RO	11 = No link	0
3		RO	10 = 10BASE-T link 01 = 100BASE-TX link 00 = 1000BASE-T link	0
2	<u>TRANSMIT</u> Indicator	RO	1 = No transmit activity. 0 = Transmit activity.	0
1	<u>RECEIVE</u> Indicator	RO	1 = No receive activity. 0 = Receive activity.	0
0	<u>QUALITY</u> Indicator	RO	1 = Quality Not OK mode. 0 = Quality OK mode.	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Status register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01000 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T LED Status register.

Slave Indicator

When bit 8 returns a 0, the device is in the Slave mode. When this bit returns a 1, the device is not in the slave mode.

FDX Indicator

When bit 7 returns a 0, the device is in the Full-Duplex mode. When this bit returns a 1, the device is not in the full-duplex mode.

INTR Indicator

When bit 6 returns a 0, the device is in the Interrupted mode. When this bit returns a 1, the device is not in the interrupted mode.

LINKSPD Indicator

When bits[4:3] return a 00, the device is in the 1000BASE-T Link mode. When these bits return a 01, the device is in the 100BASE-TX link mode. When these bits return a 10, the device is in the 10BASE-T Link mode. When these bits return an 11, the device is not linked.

Transmit Indicator

When bit 2 returns a 0, the device is in the Transmitting mode. When this bit returns a 1, the device is not in the Transmitting mode.

Receive Indicator

When bit 1 returns a 0, the device is in the Receiving mode. When this bit returns a 1, the device is not in the Receiving mode.

Quality Indicator

When bit 0 returns a 0, the device is in the Quality OK mode. When this bit returns a 1, the device is not in the Quality OK mode.

**1000BASE-T/100BASE-TX/10BASE-T LED CONTROL REGISTER (ADDRESS 1Ch,
SHADOW 09H)**

**Table 588: 1000BASE-T/100BASE-TX/10BASE-T LED Control Register
(Address 1Ch, Shadow 09h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01001 = LED Control register	0
13		R/W		1
12		R/W		0
11		R/W		0
10		R/W		1
9	Collision Blink LED Mode	R/W	1 = Blink <u>FDXLED</u> LED when a collision occurs. 0 = <u>FDXLED</u> LED indicates duplex status.	0
8	Activity Link [1]	R/W	See Description for Bit 4 below.	0
7	Reserved	R/W	Write as 0, ignore when read.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	Override Media Converter/ Remote PHY LED mode	R/W	1 = LEDs not remapped in Media Converter mode or Remote PHY mode. 0 = When in Media Converter mode or Remote PHY mode, the LEDs are mapped as follows: <ul style="list-style-type: none">• LED1: Copper link• LED2: Copper receive activity• LED3: Copper transmit activity• LED4: Copper and SerDes both linked	0
4	Activity Link [0]	R/W	Activity Link[1:0] = 11 <ul style="list-style-type: none">• 10BASE-T LINK: LED programed to <u>LINKSPD</u>[1] will be driven low.• <u>10BASE-T RX/TX</u> packets: LED programed to <u>LINKSPD</u>[1] will be driven low and high (blinking).• 100BASE-TX LINK: LED programed to <u>LINKSPD</u>[2] will be driven low.• <u>100BASE-TX RX/TX</u> packets: LED programed to <u>LINKSPD</u>[2] will be driven low and high (blinking).• 1000BASE-T LINK: LED programed to <u>ACTIVITY</u> will be driven low.• <u>1000BASE-T RX/TX</u> packets: LED programed to <u>ACTIVITY</u> will be driven low and high (blinking).	0

**Table 588: 1000BASE-T/100BASE-TX/10BASE-T LED Control Register
(Address 1Ch, Shadow 09h) (Cont.)**

Bit	Name	R/W	Description	Default
			Activity Link[1:0] = 10 <ul style="list-style-type: none"> • 10BASE-T LINK: LED programed to <u>LINKSPD[1]</u> will be driven low. • <u>10BASE-T RX/TX packets</u>: LED programed to <u>LINKSPD[1]</u> will be driven low and high (blinking). • 100BASE-TX LINK: LED programed to <u>LINKSPD[2]</u> will be driven low. • <u>100BASE-TX RX/TX packets</u>: LED programed to <u>LINKSPD[2]</u> will be driven low and high (blinking). • <u>1000BASE-T LINK</u>: LED programed to <u>LINKSPD[1]</u> and to <u>LINKSPD[2]</u> will be driven low. • <u>1000BASE-T RX/TX packets</u>: LED programed to <u>LINKSPD[1]</u> and to <u>LINKSPD[2]</u> will be driven low and high (blinking). 	
4 (Cont.)	Activity Link [0]	R/W	Activity Link[1:0] = 01 <ul style="list-style-type: none"> • 10BASE-T LINK: LED programed to <u>ACTIVITY</u> will be driven low. • 10BASE-T RX/TX packets: LED programed to <u>ACTIVITY</u> will be driven low and high (blinking). • 100BASE-TX LINK: LED programed to <u>ACTIVITY</u> will be driven low. • <u>100BASE-TX RX/TX packets</u>: LED programed to <u>ACTIVITY</u> will be driven low and high (blinking). • <u>1000BASE-T LINK</u>: LED programed to <u>ACTIVITY</u> will be driven low. • <u>1000BASE-T RX/TX packets</u>: LED programed to <u>ACTIVITY</u> will be driven low and high (blinking). 	0
			Activity Link[1:0] = 00 <ul style="list-style-type: none"> • Normal operation 	
3	ACTIVITY LED Enable	R/W	1 = Drive activity data on <u>ACTIVITY</u> LED. 0 = Drive receive data on <u>ACTIVITY</u> LED.	1
2	Remote Fault LED Enable	R/W	1 = Drive remote fault on <u>QUALITY</u> LED. 0 = Normal operation.	0
1	Link Utilization LED Selector	R/W	00 = Normal activity (fixed blink rate). 01 = Transmit activity with variable blink rate. 10 = Receive activity with variable blink rate. 11 = Transmit/receive activity with variable blink rate.	0
0			Note: This mode has higher priority than the Activity LED Enable mode in bit 3.	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Control register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01001 to enable read/write to the register address 1Ch.

Collision Blink LED Enable

Setting bit 2 =1 programs the FDLED to blink when a collision is detected.

Activity Link[1:0]

Setting bits 8 and 4 enable different LED functions. When these bits = 11, 10, and 01 the functionality of the LEDs programmed to LINKSPD[1], LINKSPD[2], and ACTIVITY are changed to the functions listed below.

Activity Link[1:0] = 11

- 10BASE-T LINK: LED programed to LINKSPD[1] will be driven low.
- 10BASE-T RX/TX packets: LED programed to LINKSPD[1] will be driven low and high (blinking).
- 100BASE-TX LINK: LED programed to LINKSPD[2] will be driven low.
- 100BASE-TX RX/TX packets: LED programed to LINKSPD[2] will be driven low and high (blinking).
- 1000BASE-T LINK: LED programed to ACTIVITY will be driven low.
- 1000BASE-T RX/TX packets: LED programed to ACTIVITY will be driven low and high (blinking).

Activity Link[1:0] = 10

- 10BASE-T LINK: LED programed to LINKSPD[1] will be driven low.
- 10BASE-T RX/TX packets: LED programed to LINKSPD[1] will be driven low and high (blinking).
- 100BASE-TX LINK: LED programed to LINKSPD[2] will be driven low.
- 100BASE-TX RX/TX packets: LED programed to LINKSPD[2] will be driven low and high (blinking).
- 1000BASE-T LINK: LED programed to LINKSPD[1] and to LINKSPD[2] will be driven low.
- 1000BASE-T RX/TX packets: LED programed to LINKSPD[1] and to LINKSPD[2] will be driven low and high (blinking).

Activity Link[1:0] = 01

- 10BASE-T LINK: LED programed to ACTIVITY will be driven low.
- 10BASE-T RX/TX packets: LED programed to ACTIVITY will be driven low and high (blinking).
- 100BASE-TX LINK: LED programed to ACTIVITY will be driven low.
- 100BASE-TX RX/TX packets: LED programed to ACTIVITY will be driven low and high (blinking).
- 1000BASE-T LINK: LED programed to ACTIVITY will be driven low.
- 1000BASE-T RX/TX packets: LED programed to ACTIVITY will be driven low and high (blinking).

Activity Link[1:0] = 00

- Normal operation

ACTIVITY LED Enable

Setting bit 3 = 1, drives activity data on the ACTIVITY LED. Otherwise, it drives receive data activity on ACTIVITY LED.

Remote Fault LED Enable

Setting bit 2 =1 drives Remote Fault on the QUALITY LED. A Remote Fault condition is detected by the NetXtreme II when bit 13 in Register 05h is set to a 1 during the auto-negotiation process

Link Utilization LED Selector

Bits[1:0] apply to the LED programmed to the Activity mode only. When in ACTIVITY LED mode, the LED expresses an estimated activity in terms of blink rate. The blink rate of the LED increases as the activity duty cycle increases by increments of 10%. For duty cycles of 0.001% to 10%, the LED blinks at 3 Hz; for duty cycles of 10% to 20%, the LED blinks at 6 Hz; and for duty cycles of 90% to 96%, the LED blinks at 30 Hz. Even though the frequency of the LED blink increases, the duty cycle of the LED stays at about 50%. The ACTIVITY LED can be programmed to display the following:

- 00 = Normal activity (fixed blink rate)
- 01 = Transmit activity with variable blink rate
- 10 = Receive activity with variable blink rate
- 11 = Transmit/receive activity with variable blink rate

**1000BASE-T/100BASE-TX/10BASE-T AUTO POWER-DOWN REGISTER (ADDRESS 1Ch,
SHADOW 0Ah)**
Table 589: Auto Power-Down Register (Address 1Ch, Shadow 0Ah)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0]. 0 = Read bits[9:0].	0
14	Shadow Register Selector	R/W	01010 = Auto Power-Down register.	0
13		R/W		1
12		R/W		0
11		R/W		1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	Reserved	R/W	Write as 0, ignore when read.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	Auto Power-Down Mode	R/W	1 = Auto Power-Down mode enabled. 0 = Auto Power-Down mode disabled.	0
4	Sleep Timer Select	R/W	1 = Sleep timer is 5.4 seconds. 0 = Sleep timer is 2.7 seconds.	0
3	Wake-up Timer Select	R/W	Counter for wake-up timer in units of 84 milliseconds.	0
2		R/W	0001 = 84 ms	0
1		R/W	0010 = 168 ms	0
0		R/W	... 1111 = 1.26 seconds	1

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] must be set to 01010 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Auto Power-Down register address 1Ch.

Auto Power-Down Mode

Setting bit 5 = 1 enables the Auto Power-Down mode. This powers-down the Copper twisted pair transmitter.

Sleep Timer Select

Setting bit 4 = 1, changes the wake-up time leaving Auto Power-Down mode.

Wake-up Timer Select

The port continues wake-up mode for a time based on the count stored in bit[3:0]. The minimum value is 84 milliseconds and the maximum value is 1.26 seconds. This only applies when the part is in Auto Power-Down mode.

SLED_1 REGISTER (ADDRESS 1Ch, SHADOW 0Bh)*Table 590: SLED_1 Register (Address 1Ch, Shadow 0Bh)*

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0]. 0 = Read bits[9:0].	0
14	Shadow Register Selector	R/W	01011 = SLED_1 register.	0
13		R/W		1
12		R/W		0
11		R/W		1
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	SLED_MUX[1]	R/W	The SLED_MUX[1] and SLED_MUX[0] bits determine how many LEDs are enabled.	0
SLED_MUX[1:0] MUX SELECT				
00 32-bit mode				
01 24-bit mode				
10 16-bit mode				
11 8-bit mode				
Note: SLED_MUX[0] can be found in register 1Ch, shadow 0Ch, bit 6.				
6	SLED TRISTATE	R/W	1 = Tristate the LED outputs. 0 = Normal operation	0
5	Reserved	R/W	Write as 0, ignore when read.	0
4	Reserved	R/W	Write as 0, ignore when read.	0
3	SLED_RESET	R/W	1 = Reset SLED Registers 0 = Normal operation	0
2	ENABLE_SLED MODE	R/W	1 = Enable SLED mode 0 = Normal operation	0
1	Reserved	R/W	Write as 0, ignore when read.	0
0	Enable SLED_IN[1]	R/W	1 = Enable SLED_IN[1] (Ball A12) as the input when in External Serial LED mode. 0 = Enable SUPER_I/SLED_IN (Ball A13) as the input when in External Serial LED mode.	0

SLED_2 REGISTER (ADDRESS 1Ch, SHADOW 0Ch)*Table 591: SLED_2 Register (Address 1Ch, Shadow 0Ch)*

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0]. 0 = Read bits[9:0].	0
14	Shadow Register Selector	R/W	01100 = SLED_2 register.	0
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	Reserved	R/W	Write as 0, ignore when read.	0
6	SLED_MUX[0]	R/W	The SLED_MUX[1] and SLED_MUX[0] bits determine how many LEDs are enabled.	0
SLED_MUX[1:0] MUX SELECT				
00 32-bit mode				
01 24-bit mode				
10 16-bit mode				
11 8-bit mode				
Note: SLED_MUX[1] can be found in register 1Ch, shadow 0Bh, bit 7.				
5	Reserved	R/W	Write as 0, ignore when read.	0
4	Reserved	R/W	Write as 0, ignore when read.	0
3	Reserved	R/W	Write as 0, ignore when read.	0
2	Reserved	R/W	Write as 0, ignore when read.	0
1	Reserved	R/W	Write as 0, ignore when read.	0
0	Reserved	R/W	Write as 0, ignore when read.	0

**1000BASE-T/100BASE-TX/10BASE-T LED SELECTOR 1 REGISTER (ADDRESS 1Ch,
SHADOW 0Dh)**

*Table 592: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register
(Address 1Ch, Shadow 0Dh)*

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01101 = LED Status register	0
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		1
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	LED2 Selector	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u>	0
6		R/W	0010 = <u>XMITLED</u>	0
5		R/W	0011 = <u>ACTIVITY</u>	0
4		R/W	0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPEED_DOWNGRADE</u> 1010 = <u>MULTICOLOR[2]</u> 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	1

**Table 592: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 Register
(Address 1Ch, Shadow 0Dh) (Cont.)**

Bit	Name	R/W	Description	Default
3	LED1 Selector	R/W	0000 = <u>LINKSPD[1]</u>	0
2		R/W	0001 = <u>LINKSPD[2]</u>	0
1		R/W	0010 = <u>XMITLED</u>	0
0		R/W	0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = <u>MULTICOLOR[1]</u> 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01101 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T LED Selector 1 register address 1Ch.

LED2 Selector

Bits[7:4] select the LED2 mode.

LED1 Selector

Bits[3:0] select the LED1 mode.

**1000BASE-T/100BASE-TX/10BASE-T LED SELECTOR 2 REGISTER (ADDRESS 1Ch,
SHADOW 0Eh)**

*Table 593: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register
(Address 1Ch, Shadow 0Eh)*

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	01110 = LED Status register	0
13		R/W		1
12		R/W		1
11		R/W		1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	LED4 Selector	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u>	0
6		R/W	0010 = <u>XMITLED</u>	1
5		R/W	0011 = <u>ACTIVITY</u>	1
4		R/W	0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = <u>MULTICOLOR[2]</u> 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0

**Table 593: 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 Register
(Address 1Ch, Shadow 0Eh) (Cont.)**

Bit	Name	R/W	Description	Default
3	LED3 Selector	R/W	0000 = <u>LINKSPD[1]</u>	0
2		R/W	0001 = <u>LINKSPD[2]</u>	0
1		R/W	0010 = <u>XMITLED</u>	1
0		R/W	0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPEED_DOWNGRADE</u> 1010 = <u>MULTICOLOR[1]</u> 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS (SGMII mode)</u> 1110 = <u>OFF (output driven high)</u> 1111 = <u>ON (output driven low)</u>	1

Write Enable

During a write to this register, setting 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 01110 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T LED Selector 2 register address 1Ch.

LED4 Selector

Bits[7:4] select the LED4 mode.

LED3 Selector

Bits[3:0] select the LED3 mode.

LED GPIO CONTROL/STATUS REGISTER (ADDRESS 1Ch, SHADOW 0Fh)

Table 594: LED GPIO Control/Status Register (Address 1Ch, Shadow Value 0Fh)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14:10	Shadow Register Selector	R/W	01111 = LED GPIO Control/Status register	01111
9:4	Reserved	R/W	Write as 00h, ignore when read.	00h
3:0	Programmable LED I/O Control	R/W	1 = Disable LED output. 0 = Enable LED output.	0h
			Bit 3: LED4 pin control Bit 2: LED3 pin control Bit 1: LED2 pin control Bit 0: LED1 pin control	

Write Enable

During a write to this register, setting LED GPIO Control/Status register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits 9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

LED GPIO Control/Status register bits[14:10] must be set to 01111 to enable read/write to the register address 1Ch.

Programmable LED I/O Control

Setting any of the bits[3:0] = 1 causes the corresponding LED pin(s) to disable the LED output(s). Clearing any of the bits[3:0] enables the corresponding LED output(s).

1000BASE-T FAST LINK DROP DETECTION REGISTER (ADDRESS 1Ch, SHADOW 10H)

Table 595: LED GPIO Control/Status Register (Address 1Ch, Shadow Value 10h)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14:10	Shadow Register Selector	R/W	10000 = 1000BASE-T Fast Link Drop Detection Register	10000
9:7	Reserved	R/W	Write as 000, ignore when read.	000
6	Link Drop Detection Selector	R/W	1 = Enable 1000BASE-T Fast Link Drop Detection 0 = Normal Operation	0
5:0	Reserved	R/W	Write as 001000, ignore when read.	001000

Write Enable

During a write to this register, setting register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits [9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

1000BASE-T Fast Link Drop Detection register bits[14:10] must be set to 10000 to enable read/write to the register address 1Ch.

Link Drop Detection Selector

Setting bit 6 = 1 enables Fast Link Drop Detection on any LED programmed to either LINKSP[1] or LINKSP[2]. Enabling Fast Link Drop Detection will reduce the time the PHY takes to indicate a link drop from 750 ms (IEEE standard) to less than 1 ms. Setting this bit has no affect in 100BASE-TX or 10BASE-T mode. When Fast Link Drop Detection is enabled, the PHY is not compliant to IEEE 802.3 Clause 28.

Bit 6 of this register determines whether the device is in IEEE-Compliant Link Drop Detection mode or in Fast Link Drop detection mode. By default the device is set to IEEE-Compliant mode. Enabling Fast Link Drop Detection will reduce the time the PHY takes to indicate a link drop from 750ms (IEEE standard) to less than 1ms. When Fast Link Drop Detection is enabled, the PHY is not compliant to IEEE 802.3 Clause 28.

Mode	1Ch, SV 10h, bit 6	Link Fail Detection Time
1000BASE-T Master	0	750 ms
1000BASE-T Master	1	1 ms
1000BASE-T Slave	0	350 ms
1000BASE-T Slave	1	1 ms
100BASE-TX	0	50 ms
100BASE-TX	1	50 ms
10BASE-T	0	100 ms
10BASE-T	1	100 ms



SGMII STATUS REGISTER (ADDRESS 1Ch, SHADOW 15h)**Table 596: SGMII Status Register (Address 1Ch, Shadow 15h)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	10101 = SGMII register	1
13		R/W		0
12		R/W		1
11		R/W		0
10		R/W		1
9	SGMII Link	RO	1 = SGMII Link up. 0 = SGMII link down.	0
8	SGMII Duplex	RO	1 = SGMII Full-Duplex. 0 = SGMII Half-Duplex.	0
7	SGMII Speed	RO	10 = SGMII 1000Mbps Speed. 01 = SGMII 100Mbps Speed. 00 = SGMII 10Mbps Speed.	0
6				0
5	SGMII Link Status Change	RO	1 = Link status change detected since last read. 0 = Link status has not change detected since last read.	0
4	Mode Select	RO	10 = SGMII to Copper Mode. 11 = Media Converter Mode.	0
3				0
2	Reserved	R/W	Write as 0, ignore on read.	0
1	Reserved	R/W	Write as 0, ignore on read.	0
0	Reserved	R/W	Write as 0, ignore on read.	0

Write Enable

During a write to this register, setting register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits 9:0, perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

SGMII register bits[14:10] must be set to 10101 to enable read/write to the register address 1Ch.

SGMII Link

If bit 9 = 1 the NetXtreme II has established either a SGMII link.

SGMII Duplex

If bit 8 = 1 the NetXtreme II is in SGMII full-duplex mode.

SGMII Speed

Bits[7:6] indicate the operating speed.

<u>Bit 7</u>	<u>Bit 6</u>	<u>Speed</u>
1	0	1000Mbps
0	1	100Mbps
0	0	10Mbps

SGMII Link Status

If bit 5 = 1 the link status has changed since the last read.

Mode Select

Bits[4:3] shows the operating mode of the NetXtreme II.

<u>Bit 4</u>	<u>Bit 3</u>	<u>Operating Mode</u>
1	0	SGMII mode
1	1	Media Converter Mode

MISC. 1000BASE-X CONTROL 2 REGISTER (ADDRESS 1Ch, SHADOW 16H)**Table 597: Misc. 1000BASE-X Control 2 Register (Address 1Ch, Shadow 16h)**

Bit	Name	R/W	Description	Default
15	Write Enable	WO	1 = Write bits [9:7] if [6] = 1 Write bits[5:0] if [6] = 0 0 = Read bits [9:0] Read will always return 0.	0
14	Shadow Register Selector	R/W	10110 = 1000BASE-X Control 2 Register	1
13		R/W	If bit 8 of this register is set prior to writing this register, then the selector value written will not be stored.	0
12		R/W		1
11		R/W	To properly read the shadow registers, one must write bit 8 as a zero, then on a subsequent write set the desired shadow register to read.	1
10		R/W		0
9	Reserved	R/W	Write as 0, ignore on read.	0
8	Reserved	R/W	Write as 0, ignore on read.	0
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Reserved	R/W	Write as 0, ignore on read.	0
6	Register Write Selector	R/W	On Writes: 1 = Write bits[9:7] when bit 15 = 1 during a write cycle. 0 = Write bits [5:0] when bit 15 = 1 during a write cycle. On Reads: 1 = 1000BASE-X link (Reg. 1Ch shadow 10101 bit 9 = 1	0
5	Enable Amplitude Signal Detect	R/W	1 = 1000BASE-X synchronization will fail if signal amplitude is not above a certain threshold. Useful in applications that do not use a fiber module and the receiver may be floating. 0 = Normal operation.	1
4	Reserved	R/W	Write as 0, ignore when read.	0
3	Filter Forced Link	R/W	1 = Synchronization status must be valid for 10 ms before link will come up when 1000BASE-X auto-negotiation is disabled. 0 = Normal operation.	1
2	Disable False Link	R/W	1 = Do not allow xmit = data when auto-negotiation is disabled unless rudi = idle detected. Force xmit = idle if rudi = config. 0 = Normal operation.	1
1	1000BASE-X Auto-Negotiation Parallel Detect Enable	R/W	1 = Turn auto-negotiation on/off in order to link up with link partner. Algorithm based on received code words. 0 = Disable parallel detection.	1

Table 597: Misc. 1000BASE-X Control 2 Register (Address 1Ch, Shadow 16h) (Cont.)

Bit	Name	R/W	Description	Default
0	FIFO ELASTICITY[1] (MSB) SGMII Transmit and Receive	R/W	FIFO ELASTICITY [MSB: LSB] 11 = Supports 18 KB packets. 10 = Supports 13.5 KB packets. 01 = Supports 9 KB packets (default value). 00 = Supports 4.5 KB packets. LSB located at register 1Ch, shadow 1Bh, bit 1.	0

Write Enable

When bit 15 = 1 and bit 6 = 1 of this register, bits[9:7] can be written.

When bit 15 = 1 and bit 6 = 0 of this register, bits[5:0] can be written.

A read will always return a 0.

Shadow Register Selector

To properly read this register, bit 8 must be written as a 0. Then write to bits[14:10] = 10110 to read this register.

Enable Amplitude Signal Detect

When bit 5 = 1, the 1000BASE-X synchronization will fail if signal amplitude is not above a certain threshold. Useful in applications that do not use a fiber module and the receiver may be floating.

Filter Forced Link

When bit 3 = 1, Synchronization status must be valid for 10 ms before link will come up when 1000BASE-X auto-negotiation is disabled.

Disable False Link

Setting bit 2 = 1 will not allow xmit = data when auto-negotiation is disabled unless rudi = idle detected. Force xmit = idle if rudi = config.

1000BASE-X Auto-Negotiation Parallel Detect Enable

Setting bit 1 = 1 will turn auto-negotiation on and off in order to link up with link partner. The algorithm is based on received code words.

FIFO ELASTICITY[1]

There are two bits that control the transmit and receive packet size when in 10/100 SGMII mode.

- FIFO ELASTICITY[1] is the MSB and is located in register 1Ch, shadow 16h, bit 0.
- FIFO ELASTICITY[0] is the LSB and is located in register 1Ch, shadow 1Bh, bit 1.

The default for FIFO ELASTICITY[1] is 0, and the default value for FIFO ELASTICITY[0] is 1.

1000BASE-T/100BASE-TX/10BASE-T AUTODETECT SGMII/MEDIA CONVERTER REGISTER (ADDRESS 1Ch, SHADOW 18H)

Table 598: 1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow 18h)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11000 = 1000BASE-T/100BASE-TX/10BASE-T	1
13		R/W	Autodetect SGMII/Media Converter register	1
12		R/W		0
11		R/W		0
10		R/W		0
9	Reserved	RO	Ignore on read.	0
8	SGMII Transmit = Data	RO	1 = SGMII interface is in xmit = data state. 0 = SGMII interface is in the xmit = config or idle state.	0
7	Reserved	RO	Ignore on read.	0
6	Reserved	RO	Ignore on read.	0
5	Reserved	RO	Ignore on read.	0
4	Reserved	R/W	Write as 0, ignore on read.	0
3	Media Converter Mode	R/W	1 = Do not allow copper side to link up until Fiber Signal Detect/RX_LOS is set and SerDes synchronization is valid. 0 = Normal Media Converter Mode Note: This bit is only valid if Register 1Ch, shadow 1Fh, bits[2:1] = 11 and Register 1Ch, shadow 1Eh, bit 0 = 0.	0
2	Reserved	R/W	Write as 0, ignore on read.	0
1	SGMII 10/100 Low Receive Latency	R/W	1 = 1 Bypass the RX FIFO when in 10BASE-T or 100BASE-TX mode to reduce latency. Note: This should only be done if the NetXtreme II and the switch reference clocks are run off the same reference source	0
0	Auto-Detect SGMII/Media Converter Mode	R/W	1 = Enable auto-detection between SGMII and Media Converter modes. 0 = Normal operation.	0

Write Enable

During a write to this register, setting bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] of this register must be set to 11000 to enable read/write to the 1000BASE-T/100BASE-TX/10BASE-T Autodetect SGMII/Media Converter address 1Ch.

SGMII Transmit = Data

When bit 8 = 1, the NetXtreme II is able to transmitting data on the SGMII interface. When bit 8 = 0, the NetXtreme II is in the xmit = config or idle state.

Media Converter Mode

When bit 3 = 1, the 1000BASE-T interface does not link up until the 1000BASE-X interface is linked. When bit 3 = 0 the 1000BASE-X interface does not link up until the 1000BASE-T interface is linked.

Note: This bit is only valid if Register 1Ch, shadow 1Fh, bits[2:1] = 11 and Register 1Ch, shadow 1Eh, bit 0 = 0.

SGMII 10/100 Low Receive Latency

Setting bit 1= 1 bypasses the RX FIFO when in 10BASE-T or 100BASE-TX mode to reduce latency. This should only be done if the NetXtreme II and the switch reference clocks are run off the same reference source.

Auto-Detect SGMII/Media Converter Mode

Setting bit 0 = 1 enables the NetXtreme II to auto-detection between SGMII and Media Converter modes.

This bit must be set = 0 when changing the contents of register 1Ch, shadow 1Fh, bits[2:1].

AUXILIARY CONTROL REGISTER (ADDRESS 1Ch, SHADOW 1Bh)

The following is enabled by register 1Ch with shadow in bits[14:10] = 11011.

Table 599: Auxiliary Control Register (Address 1Ch, Shadow 1Bh)

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11011 = Auxiliary Control register	1
13		R/W		1
12		R/W		0
11		R/W		1
10		R/W		1
9	Use SerDes Mode Counters	R/W	1 = Use registers 12h to 14h for SerDes data. 0 = Use registers 12h to 14h for copper data.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	SGMII/ False Carrier Mode	R/W	1 = Send packets with TXEN, TXER, TXD = 55h for duration of false carrier in SGMII Half-Duplex mode or GBIC Half-Duplex mode. 0 = Ignore false carriers in SGMII or GBIC mode.	1
6	Disable Carrier Extend	R/W	1 = Force RXER and RXD to zeros in TRR+ extend state (PCS receive state). 0 = Normal operation.	0
5	Disable TRRR	R/W	1 = Bypass extend_by_1 state (PCS transmit state). 0 = Normal operation.	0
4	Reserved	R/W	Write as 0, ignore when read.	0
3	Reserved	R/W	Write as 0, ignore when read.	0
2	Reserved	R/W	Write as 1, ignore when read.	1
1	FIFO ELASTICITY[0] (LSB) SGMII Transmit and Receive	R/W	FIFO ELASTICITY [MSB:LSB] 11 = Supports 18 KB packets. 10 = Supports 13.5 KB packets. 01 = Supports 9 KB packets (default value). 00 = Supports 4.5 KB packets. Note: MSB located at register 1Ch, shadow 16h, bit 0.	1
0	Disable Receive CRC Checker	R/W	1 = Disable CRC checker. 0 = Enable CRC checker.	1

Write Enable

During a write to this register, setting Auxiliary Control register bit 15 to a 1 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 set to a 0 and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Bits[14:10] of this register must be set to 11011 to enable read/write to the Auxiliary Control register.

Use SerDes Mode Counters

When bit 9 = 1, registers 12h to 14h are used for SerDes Data. When bit 9 = 0, registers 12h to 14h are used for Copper Data.

SGMII False Carrier Mode

Setting bit 7 = 1 causes the data 55h to be sent out for the duration of a false carrier event when in SGMII Half-Duplex mode.

Disable Carrier Extend

Setting bit 6 = 1 causes the carrier extend symbol to be replaced with zeros.

Disable TRRR

Setting bit 5 = 1 causes the PHY to transmit only a TRR for odd-sized packets.

FIFO ELASTICITY[1]

There are two bits that control the transmit and receive packet size.

- FIFO ELASTICITY[1] is the MSB and is located in register 1Ch, shadow 16h, bit 0.
- FIFO ELASTICITY[0] is the LSB and is located in register 1Ch, shadow 1Bh, bit 1.

The default for FIFO ELASTICITY[1] is 0, and the default value for FIFO ELASTICITY[0] is 1.

Disable Receive CRC Checker

Setting bit 0 = 1 disables the Receive CRC checker.

AUXILIARY SGMII STATUS REGISTER (ADDRESS 1Ch, SHADOW 1Ch)**Table 600: Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow 1Ch)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11100 = Auxiliary SGMII register	1
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		0
9	Link Status Change	RO	1 = Link status change has occurred since last read.	0
		LH	0 = Link status change has not occurred since last read.	
8	SGMII Selector Mismatch	RO	1 = SGMII selector mismatch in SGMII mode. 0 = Fiber, Copper, GBIC mode, or SGMII selector does not mismatch, or auto-negotiation is disabled.	0
7	Auto-Negotiation Resolution Error	RO	1 = Auto-negotiation HCD is none (no common half-duplex or full-duplex abilities). 0 = SGMII mode, or auto-negotiation disabled, or no resolution error.	0
6	Reserved	RO	Ignore on read.	0
5	Reserved	RO	Ignore on read.	0
4	Auto-Negotiation Page Received	RO	1 = Page has been received since last read. LH 0 = Page has not been received since last read.	0
3	Current Operating Duplex Mode	RO	1 = Phy is operating in full-duplex mode. 0 = Phy is operating in half-duplex mode (or auto-negotiation has not completed).	0
2	SGMII Link Status	RO	1 = Link is up on SGMII side. 0 = Link is down on SGMII side.	0
1	Reserved	RO	Ignore on read.	0
0	Reserved	RO	Ignore on read.	0

Write Enable. During a write to this register, setting Auxiliary SGMII Status register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector. Bits[14:10] of this register must be set to 11100 to enable read/write to the Auxiliary SGMII Status register

Link Status Change. Bit 9 = 1 indicates that the link status has changed since the last register read.

SGMII Selector Mismatch. Bit 8 = 1 indicates an SGMII selector mismatch in SGMII mode.

Auto-Negotiation Resolution Error. Bit 7 = 1 indicates auto-negotiation HCD is none (no common half-duplex or full-duplex abilities).

Auto-Negotiation Page Received. Bit 4 = 1 indicates auto-negotiation page has been received since last read.

Current Operating Duplex Mode. Bit 3 = 1 indicates the PHY is operating in full-duplex mode.

SGMII Link Status. Bit 2 = 1 indicates the PHY link is up on the SGMII side.

MISC. SGMII STATUS REGISTER (ADDRESS 1Ch, SHADOW 1Dh)**Table 601: Misc. 1000BASE-X Status Register (Address 1Ch, Shadow 1Dh)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0] 0 = Read bits[9:0]	0
14	Shadow Register Selector	R/W	11101 = Misc. SGMII register	1
13		R/W		1
12		R/W		1
11		R/W		0
10		R/W		1
9	TX FIFO Error	RO LH	1 = Transmit FIFO error since last read. 0 = No transmit FIFO error since last read.	0
8	RX FIFO Error	RO LH	1 = Receive FIFO error since last read. 0 = No receive FIFO error since last read.	0
7	Reserved	RO LH	Ignore on read.	0
6	Reserved	RO LH	Ignore on read.	0
5	False Carrier Detected on the SGIN± input	RO LH	1 = False carrier detected since last read. 0 = No false carriers detected since last read.	0
4	Receive CRC Error Detected on the SGIN± input	RO LH	1 = Receive CRC Error detected since last read. 0 = No Receive CRC error detected since last read or mode is disabled via register 1Ch, shadow 1Bh, bit 0.	0
3	Transmit Error Detected on the SGOUT± output	RO LH	1 = Transmit error code detected since last read (rx_data_error state in PCS receive). 0 = No transmit error code detected since last read.	0
2	Receive Error Detected on the SGIN± input	RO LH	1 = Receive error since last read. (early_end state in PCS receive). 0 = No receive error since last read.	0
1	Carrier Extend Error Detected on the SGIN± input	RO LH	1 = Carrier extend error since last read (extend_err state in PCS receive). 0 = No carrier extend error since last read.	0
0	Early End Extension Detected on the SGIN± input	RO LH	1 = Early end extension since last read (early_end_ext state in PCS receive). 0 = No early end extension since last read.	0

Write Enable. During a write to this register, setting Misc. SGMII Status register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow values in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector. Register bits[14:10] must be set to 11101 to enable read/write to the Misc. 1000BASE-X Status register.

TX FIFO Error. Bit 9 = 1 indicates a transmit FIFO error since the last read.

RX FIFO Error. Bit 8 = 1 indicates a receive FIFO error since the last read.

False Carrier Detected. Bit 5 = 1 indicates a false carrier detected since the last read.

CRC Error Detected. Bit 4 = 1 indicates a CRC error detected since the last read.

Transmit Error Detected. Bit 3 = 1 indicates a transmit error code detected since the last read.

Receive Error Detected. Bit 2 = 1 indicates a receive error code detected since the last read.

Carrier Extend Error Detected. Bit 1 = 1 indicates a carrier extend error since the last read.

Early End Extension Detected. Bit 0 = 1 indicates an early end extension since the last read.

MODE CONTROL REGISTER (ADDRESS 1Ch, SHADOW 1Fh)

**Table 602: Mode Control Register
(Address 1Ch, Shadow 1Fh)**

Bit	Name	R/W	Description	Default
15	Write Enable	R/W	1 = Write bits[9:0]. 0 = Read bits[9:0].	0
14	Shadow Register Selector	R/W	11111 = LED Status register.	1
13		R/W		1
12		R/W		1
11		R/W		1
10		R/W		1
9	Reserved	RO	Ignore on read.	0
8	Mode Select Change	RO LH	1 = Interface Mode Select status changed since last read. 0 = Interface Mode Select status did not change since last read.	0
7	Copper Link	RO	1 = Link is good on the copper interface. 0 = Copper link is down.	0
6	SGMII Link	RO	1 = Link is good on the SGMII interface. 0 = SGMII link is down.	0
5	Copper Energy Detect	RO	1 = Energy detected on the copper interface. 0 = Energy not detected on the copper interface.	0
4	Reserved	RO	Ignore on read.	0
3	Reserved	RO	Ignore on read.	1
2	Mode Select	R/W	10 = SGMII 11 = Media Converter	1
1				0
0	Enable SGMII Registers	R/W	1 = Select SGMII registers for addresses 00h–0Fh. 0 = Select copper registers for addresses 00h–0Fh.	0

Write Enable

During a write to this register, setting Mode Control register bit 15 allows writing to bits[9:0] of this register. For reading the values of bits[9:0], perform an MDIO write with bit 15 cleared and preferred shadow in bits[14:10]. The next MDIO read of register address 1Ch contains the preferred shadow register values in bits[9:0].

Shadow Register Selector

Register bits[14:10] of this register must be set to 11111 to enable read/write to the Mode Control register address 1Ch.

Mode Select Change

Bit 8 = 1 indicates there is change in the interface mode selection; otherwise, it reads a 0.

Copper Link

Bit 7 = 1 indicates the link status of the copper interface is up; otherwise, it reads a 0.

SGMII Link

When in bit 6 = 1 indicates the SGMII interface is linked.

Copper Energy Detect

Bit 5 = 1 indicates that energy (link pulses, FLPs, or data) is detected on the copper interface.

Mode Select

When read, bits[2:1] give the mode the NetXtreme II is in. Writing to bits[2:1] sets the NetXtreme II to the following modes.

10 = SGMII

11 = Media Converter

Enable SGMII Registers

Setting bit 0 = 1 enables the SGMII register set for addresses 00h–0Fh.

Setting bit 0 = 0 enables the copper register set for addresses 00h–0Fh.

1000BASE-T/100BASE-TX/10BASE-T MASTER/SLAVE SEED REGISTER (ADDRESS 1DH) BIT 15 = 0

Table 603: 1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed Register (Address 1Dh) Bit 15 = 0

Bit	Name	R/W	Description	Default
15	Enable Shadow Register	R/W	1 = Select Shadow register. 0 = Normal operation. Writes to the selected register are done on a single cycle.	0
14	Master/Slave Seed Match	RO LH	1 = Seeds match. 0 = Seeds do not match.	0
13	Link Partner Repeater/DTE Bit	RO	1 = Link partner is a repeater/switch device. 0 = Link partner is a DTE device.	0
12	Link Partner Manual Master/ Slave Configuration Value	RO	1 = Link partner is configured as master. 0 = Link partner is configured as slave.	0
11	Link Partner Manual Master/ Slave Configuration Enable	RO	1 = Link partner manual master/slave configuration enabled. 0 = Link partner manual master/slave configuration disabled.	0
10	Local Master/Slave Seed Value	R/W	Returns the automatically generated master/slave random seed.	0
9		R/W		0
8		R/W		0
7		R/W		0
6		R/W		0
5		R/W		0
4		R/W		0
3		R/W		0
2		R/W		0
1		R/W		0
0		R/W		0

Enable Shadow Register

When bit 15 = 0, the Master/Slave Seed register is selected. If bit 15 = 1, the shadow register HCD Status register is selected for read/write.

Master/Slave Seed Match

When bit 14 returns a 1, the master/slave seed matches; otherwise, it returns a 0.

Link Partner Repeater/DTE Bit

When bit 13 returns a 1, it indicates the link partner is configured as a repeater or a switch. If this bit returns a 0, it indicates the link partner is configured as a DTE port.

Link Partner Manual Master/Slave Configuration Value

When bit 12 returns a 1, it indicates the link partner is configured as a master. If this bit returns a 0, it indicates the link partner is configured as a slave.

Link Partner Manual Master/Slave Configuration Enable

When bit 11 returns a 1, it indicates the link partner manual master/slave configuration is enabled. If this bit returns a 0, the link partner manual master/slave configuration is disabled.

Local Master/Slave Seed Value

Bits[10:0] return the automatically generated local master/slave seed value.

1000BASE-T/100BASE-TX/10BASE-T HCD STATUS REGISTER (ADDRESS 1DH) BIT 15 = 1

Table 604: 1000BASE-T/100BASE-TX/10BASE-T HCD Status Register (Address 1Dh) Bit 15 = 1

Bit	Name	R/W	Description	Default
15	Enable Shadow Register	R/W	1 = Select Shadow register. 0 = Normal operation.	0
14	Ethernet@Wirespeed Disable Gigabit Advertising	RO	1 = Disable advertising Gigabit. 0 = Advertise Gigabit based on register 09h.	0
13	Ethernet@Wirespeed Disable 100TX Advertising	RO	1 = Disable advertising 100TX. 0 = Advertise 100TX based on register 04h.	0
12	Ethernet@Wirespeed Downgrade	RO LH	1 = Ethernet@Wirespeed downgrade occurred since last read. 0 = Ethernet@Wirespeed downgrade cleared.	0
11	HCD 1000BASE-T Full-Duplex	RO LH	1 = Gigabit full-duplex occurred since last read. 0 = HCD cleared.	0
10	HCD 1000BASE-T Half-Duplex	RO LH	1 = Gigabit half-duplex occurred since last read. 0 = HCD cleared.	0
9	HCD 100BASE-TX Full-Duplex	RO LH	1 = 100BASE-TX full-duplex occurred since last read. 0 = HCD cleared.	0
8	HCD 100BASE-TX Half-Duplex	RO LH	1 = 100BASE-TX half-duplex occurred since last read. 0 = HCD cleared.	0
7	HCD 10BASE-T Full-Duplex	RO LH	1 = 10BASE-T full-duplex occurred since last read. 0 = HCD Cleared.	0
6	HCD 10BASE-T Half-Duplex	RO LH	1 = 10BASE-T half-duplex occurred since last read. 0 = HCD cleared.	0
5	HCD 1000BASE-T Full-Duplex (Link Never Came Up)	RO LH	1 = Gigabit full-duplex HCD and Link Never Came Up occurred since the last read. 0 = HCD cleared.	0
4	HCD 1000BASE-T Half-Duplex (Link Never Came Up)	RO LH	1 = Gigabit half-duplex HCD and Link Never Came Up occurred since the last read. 0 = HCD cleared.	0
3	HCD 100BASE-TX Full-Duplex (Link Never Came Up)	RO LH	1 = 100BASE-TX full-duplex HCD and Link Never Came Up occurred since the last read. 0 = HCD cleared.	0
2	HCD 100BASE-TX Half-Duplex (Link Never Came Up)	RO LH	1 = 100BASE-TX half-duplex HCD and Link Never Came Up occurred since the last read. 0 = HCD cleared.	0
1	HCD 10BASE-T Full-Duplex (Link Never Came Up)	RO LH	1 = 10BASE-T full-duplex HCD and Link Never Came Up occurred since the last read. 0 = HCD cleared.	0
0	HCD 10BASE-T Half-Duplex (Link Never Came Up)	RO LH	1 = 10BASE-T half-duplex HCD and Link Never Came Up occurred since the last read. 0 = HCD cleared.	0



Note: Bits[12:0] are also cleared when auto-negotiation is disabled via MII register 00h, bit 12 = 1; or restarted via MII register 00h, bit 9 = 1.

Enable Shadow Register

When bit 15 = 0, the 1000BASE-T/100BASE-TX/10BASE-T Master/Slave Seed register is selected. When bit 15 = 1, the HCD Status register is selected for read/write.

Ethernet@Wirespeed Disable Gigabit Advertising

When bit 14 = 1, 1000BASE-T Half-Duplex and 1000BASE-T Full-Duplex are not advertised.

Ethernet@Wirespeed Disable 100BASE-TX Advertising

When bit 13 = 1, 100BASE-TX Half-Duplex and 100BASE-TX Full-Duplex are not advertised.

Ethernet@Wirespeed Downgrade

When bit 12 = 1, an Ethernet@Wirespeed downgrade has occurred since the last read.

HCD 1000BASE-T FDX

When bit 11 returns a 1, it indicates that a Gigabit Full-Duplex HCD has occurred since the last read.

HCD 1000BASE-T

When bit 10 returns a 1, it indicates that a Gigabit Half-Duplex HCD has occurred since the last read.

HCD 100BASE-TX FDX

When bit 9 returns a 1, it indicates that a 100BASE-TX Full-C HCD has occurred since the last read.

HCD 100BASE-TX

When bit 8 returns a 1, it indicates that a 100BASE-TX Half-Duplex HCD has occurred since the last read.

HCD 10BASE-T FDX

When bit 7 returns a 1, it indicates that a 10BASE-T Full-Duplex HCD has occurred since the last read.

HCD 10BASE-T

When bit 6 returns a 1, it indicates that a 10BASE-T Half-Duplex HCD has occurred since the last read.

HCD 1000BASE-T FDX (Link Never Came Up)

When bit 5 returns a 1, it indicates that a Gigabit Full-Duplex HCD has occurred, but the link was not established since the last read.

HCD 1000BASE-T (Link Never Came Up)

When bit 4 returns a 1, it indicates that a Gigabit Half-Duplex HCD has occurred, but the link was not established since the last read.

HCD 100BASE-TX FDX (Link Never Came Up)

When bit 3 returns a 1, it indicates that a 100BASE-TX Full-Duplex HCD has occurred, but the link was not established since the last read.

HCD 100BASE-TX (Link Never Came Up)

When bit 2 returns a 1, it indicates that a 100BASE-TX Half-Duplex HCD has occurred, but the link was not established since the last read.

HCD 10BASE-T FDX (Link Never Came Up)

When bit 1 returns a 1, it indicates that a 10BASE-T Full-Duplex HCD has occurred, but the link was not established since the last read.

HCD 10BASE-T FDX (Link Never Came Up)

When bit 0 returns a 1, it indicates that a 10BASE-T Half-Duplex HCD has occurred, but the link was not established since the last read.

TEST REGISTER 1 (ADDRESS 1Eh)*Table 605: Test Register 1 (Address 1Eh)*

Bit	Name	R/W	Description	Default
15	CRC Error Counter Selector	R/W	1 = Receiver NOT_OK Counters (register 14h) becomes 16-bit CRC error counter (CRC errors are counted only after this bit is set). 0 = Normal operation.	0
14	Transmit Error Code Visibility	R/W	1 = False Carrier Sense Counters (register 13h) counts packets received with transmit error codes. 0 = Normal operation.	0
13	Reserved	R/W	Write as 0, ignore when read.	0
12	Force Link 10/100/1000BASE-T	R/W	1 = Force Link State machine into link pass state. 0 = Normal operation.	0
11	Reserved	R/W	Write as 0, ignore when read.	0
10	Reserved	R/W	Write as 0, ignore when read.	0
9	Reserved	R/W	Write as 0, ignore when read.	0
8	Reserved	R/W	Write as 0, ignore when read.	0
7	Manual Swap MDI State	R/W	1 = Manually Swap MDI state. 0 = Normal operation.	0
6	Reserved	R/W	Write as 0, ignore when read.	0
5	Reserved	R/W	Write as 0, ignore when read.	0
4	Reserved	R/W	Write as 0, ignore when read.	0
3	Reserved	R/W	Write as 0, ignore when read.	0
2	Reserved	R/W	Write as 0, ignore when read.	0
1	Reserved	R/W	Write as 0, ignore when read.	0
0	Reserved	R/W	Write as 0, ignore when read.	0

CRC Error Counter Selector

Setting bit 15 = 1 enables the counting register 14h to count and store the number of CRC errors.

Packets Received with Transmit Error Codes Counter Selector

Setting bit 14 = 1 enables register 13h to start counting packets with transmit error codes and store the counts in register 13h.

Force Link

Setting bit 12 = 1 forces the link state machine into the Link Pass state.

Manual Swap MDI State

Setting bit 7 = 1 manually swaps the MDI transmit and receive pairs during forced 100BASE-TX and 10BASE-T operation. When this bit is set, the NetXtreme II transceiver transmits on pairs TRD[1]{8:1}± and receives on TRD[0]{8:1}± when operating in 100BASE-TX and 10BASE-T modes. If this bit is cleared, the NetXtreme II transmits on pairs TRD[0]{8:1}± and receives on TRD[0]{8:1}± when operating in 100BASE-TX and 10BASE-T modes. This bit is ignored when auto-negotiation is enabled.



EXPANSION REGISTERS

EXPANSION REGISTER 00H: RECEIVE/TRANSMIT PACKET COUNTER REGISTER (ADDRESS 15H)

Expansion register 00h is enabled by writing to “Expansion Register Access Register (Address 17h)” bits[11:0] = ‘F00’h, and read/write access is through register 15h.

Table 606: Expansion Register 00h: Receive/Transmit Packet Counter Register (Address 15h)

Bit	Name	R/W	Description	Default
15:0	Packet Counter (Copper Only)	R/W CR	Returns the transmitted and received packet count.	0000h

Packet Counter (Copper Only)

The mode of this counter is set by bit 11 of “[1000BASE-T/100BASE-TX/10BASE-T Miscellaneous Control Register \(Address 18h, Shadow 7h\)](#)”. When bit 11 =1, then receive packets (both good and bad CRC error packets) are counted. When bit 11 = 0, then transmit packets (both good and bad CRC error packets) are counted. This counter is cleared on a read and freezes at FFFFh.

EXPANSION REGISTER 04H: MULTICOLOR LED SELECTOR REGISTER (ADDRESS 15H)

Expansion register 04h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F04’h, and read/write access is through register 15h.

Table 607: Expansion Register 04h: Multicolor LED Selector Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Reserved	R/W	Write as 0, ignore on read.	0
10	Reserved	R/W	Write as 0, ignore on read.	0
9	Flash Now	R/W	1 = Initiate a multicolor LED flash. SC Note: This works only when the multicolor selector is set to 0111.	0
8	In Phase	R/W	1 = <u>MULTICOLOR[1]</u> and <u>MULTICOLOR[2]</u> are in phase. 0 = <u>MULTICOLOR[1]</u> and <u>MULTICOLOR[2]</u> are in opposite phase. Note: This is only valid when Multicolor LED Selector bits are set to 0000, 0010, 0011, 0110, 0111, 1000, 1001, and 1010.	0
7	MULTICOLOR[2] LED Selector	R/W	Selects the multicolor mode for MULTICOLOR[2] LED.	0
6		R/W	0000 = Encoded link/activity LED.	0
5		R/W	0001 = Encoded speed LED.	0
4		R/W	0010 = Activity flash LED. 0011 = Full-duplex LED. 0100 = Forced off. 0101 = Forced on. 0110 = Alternating LED (50% duty cycle with a 320 ms period). 0111 = Flashing LED (toggling between two of the states with an 80 ms period). 1000 = Link LED. 1001 = Activity LED. 1010 = Programmable blink LED.	0
3	MULTICOLOR[1] LED Selector	R/W	Selects the multicolor mode for MULTICOLOR[1] LED.	0
2		R/W	0000 = Encoded link/activity LED.	0
1		R/W	0001 = Encoded speed LED.	0
0		R/W	0010 = Activity flash LED. 0011 = Full-duplex LED. 0100 = Forced off. 0101 = Forced on. 0110 = Alternating LED (50% duty cycle with a 320 ms period). 0111 = Flashing LED (toggling between two of the states with an 80 ms period). 1000 = Link LED. 1001 = Activity LED. 1010 = Programmable blink LED.	0

Flash Now

Asserting this bit causes a single flash to occur on either MULTICOLOR[2:1] LEDs, as long as its multicolor selector is set to 0111.

In Phase

When both LEDs are selected to the same mode, the MULTICOLOR[2:1] output pins toggle at the same time. This bit determines whether the pins are identical to each other or inverses of each other. When the two LED pins are attached to a special multicolored LED, the resulting LED colors alternate either between off/amber (in phase) or red/green (out of phase).

MULTICOLOR[2] LED Selector

Bits[7:4] select the multicolor LED mode for MULTICOLOR[2]. It is up to the user to determine what functions should appear on the two LED pins. For example, if the user wants a different color toggling operation rather than the operation mentioned above, such as red/amber, the user can put one of the selectors to the desired toggle mode, and other selector to *forced on*.

MULTICOLOR[1] LED Selector

Bits[3:0] select the multicolor LED mode for MULTICOLOR[1].

EXPANSION REGISTER 05H: MULTICOLOR LED FLASH RATE CONTROLS REGISTER (ADDRESS 15H)

Expansion register 05h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F05’h, and read/write access is through register 15h.

Table 608: Expansion Register 05h: Multicolor LED Flash Rate Controls Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Alternating Rate	R/W	Determines the width and gap for multicolor LED selector 0110 (alternating LED mode).	0
10		R/W	00h = 21-ms width, 21-ms gap.	0
9		R/W	01h = 42-ms width, 42-ms gap.	0
8		R/W	02h = 63-ms width, 63-ms gap.	1
7		R/W	...	1
6		R/W	07h = 168-ms width, 168-ms gap. ... 3Fh = 1.344 seconds.	1
5	Flash Rate	R/W	Determines the width and minimum gap of every flash pulse for multicolor LED selector 0000 (encoded link/activity mode), 0010 (activity flash mode), and 0111 (flashing LED mode).	0
4		R/W	00h = 21-ms width.	0
3		R/W	01h = 42-ms width.	0
2		R/W	02h = 63-ms width.	0
1		R/W	...	0
0		R/W	3Fh = 1.344 seconds.	1

Alternating Rate

Setting bits[11:6] changes the width and gap of the alternating LED modes. These bits are only valid when the MULTICOLOR[1] LED Selector and/or the MULTICOLOR[2] LED Selector bits = 0110. The duty cycle of the LED is exactly 50%.

Flash Rate

Setting bits[5:0] determines the width and minimum gap of the flashing pulse. These bits are only valid when the MULTICOLOR[1] LED Selector and/or the MULTICOLOR[2] LED Selector bits = 0111. The duty cycle of the flash rate is not exactly 50%.

EXPANSION REGISTER 06H: MULTICOLOR LED PROGRAMMABLE BLINK CONTROLS REGISTER (ADDRESS 15H)

Expansion register 06h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F06’h, and read/write access is through register 15h.

Table 609: Expansion Register 06h: Multicolor LED Programmable Blink Controls Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Reserved	R/W	Write as 0, ignore on read.	0
10	Reserved	R/W	Write as 0, ignore on read.	0
9	Reserved	R/W	Write as 0, ignore on read.	0
8	Reserved	R/W	Write as 0, ignore on read.	0
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Reserved	R/W	Write as 0, ignore on read.	0
5	Blink Update Now	R/W	1 = Change to the new blink rate now. 0 = Wait 1 second before changing the blink rate. Controls when a change in the blink rate is actually displayed on the Programmable Blink LED.	0
4	Blink Rate	R/W	Programs the number of blinks per second of the Programmable Blink LED.	0
3		R/W	00000 = No blink.	0
2		R/W	00001 = 1 blink per second.	0
1		R/W	00010 = 2 blinks per second.	0
0		R/W	00011 = 3 blinks per second. ... 11111 = 31 blinks per second.	0

Blink Update Now

Setting bit 5 = 1 immediately updates the blink rate. Clearing this bit causes the blink rate to be updated after the 1-second interval timer expires. This bit is only valid when the MULTICOLOR[1] LED Selector and/or the MULTICOLOR[2] LED Selector bits = 1010.

Blink Rate

Setting bits[4:0] determines the blink rate of the Programmable Blink LED. These bits are only valid when the MULTICOLOR[1] LED Selector or the MULTICOLOR[2] LED Selector bits = 0000, 0010, or 0111.

EXPANSION REGISTER 07H: 100BASE-FX FAR END FAULT REGISTER (ADDRESS 15H)

Expansion register 07h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F07’h, and read/write access is through register 15h.

Table 610: Expansion Register 07h: 100BASE-FX Far End Fault Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Reserved	R/W	Write as 0, ignore on read.	0
14	Reserved	R/W	Write as 0, ignore on read.	0
13	Reserved	R/W	Write as 0, ignore on read.	0
12	Reserved	R/W	Write as 0, ignore on read.	0
11	Reserved	R/W	Write as 0, ignore on read.	0
10	Reserved	R/W	Write as 0, ignore on read.	0
9	Reserved	R/W	Write as 0, ignore on read.	0
8	Reserved	R/W	Write as 0, ignore on read.	0
7	Reserved	R/W	Write as 0, ignore on read.	0
6	Reserved	R/W	Write as 0, ignore on read.	0
5	Reserved	R/W	Write as 0, ignore on read.	0
4	Reserved	R/W	Write as 0, ignore on read.	0
3	Reserved	R/W	Write as 0, ignore on read.	0
2	100BASE-FX SD Select	R/W	1 = Do not use SD/RX_LOS input. 0 = Use SD/RX_LOS input.	
1	Faulting	R/O	1 = Far End indication detected. 0 = No Far End indication detected.	0
0	Far End Fault Enable	R/W	1 = Enable Far End Fault. 0 = Disable Far End Fault.	0

100BASE-FX SD Select

Setting bit 2 = 1 allows the NetXtreme II to establish a 100BASE-FX link when there is no SD or RX_LOS input from the optical module to the NetXtreme II. Link is established through the exchange of idles. When bit 2 = 0, the NetXtreme II does not establish a link unless the SD or RX_LOS input is active and it is receiving valid idle signals from the optical module.

Faulting

Bit 1 = 1 indicates that a Far End Fault has been detected.

Far End Fault Enable

Setting bit 0 = 1, enables the Far End Fault feature per the IEEE 802.3 section 24.3.2.1 Far-End fault.

EXPANSION REGISTER 09H: CHANNEL SWAPPING (MDI REVERSE) REGISTER (ADDRESS 15H)

Expansion register 09h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F09’h, and read/write access is through register 15h.

Table 611: Expansion Register 09h: Channel Swapping (MDI Reverse) Register (Address 15h)

Bit	Name	R/W	Description	Default
15:9	Reserved	R/W	Write as 0, ignore on read.	00h
8	Enable Channel Swapping	R/W	1 = Enable channel swapping using bits[7:0]. 0 = Disable channel swapping, ignore bits[7:0]	0
7:6	TRD[3] (Channel 3 Select)	R/W	Selects which channel will map to Channel 3. 11 = Channel 3 remains Channel 3. 10 = Reserved. 01 = Reserved. 00 = Channel 3 mapped to Channel 0.	11
5:4	TRD[2] (Channel 2 Select)	R/W	Selects which channel will map to Channel 2. 11 = Reserved. 10 = Channel 2 remains Channel 2. 01 = Channel 2 mapped to Channel 1. 00 = Reserved.	10
3:2	TRD[1] (Channel 1 Select)	R/W	Selects which channel will map to Channel 1. 11 = Reserved. 10 = Channel 1 mapped to Channel 2. 01 = Channel 1 remains Channel 1. 00 = Reserved.	01
1:0	TRD[0] (Channel 0 Select)	R/W	Selects which channel will map to Channel 0. 11 = Channel 0 mapped to Channel 3 10 = Reserved. 01 = Reserved. 00 = Channel 0 remains Channel 0.	00

Note: User must be careful to avoid over-mapping channels. Logic will not automatically resolve conflicts.

EXPANSION REGISTER 0Bh: PORT INTERRUPT STATUS REGISTER (ADDRESS 15H)

Expansion register 0Bh is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits 11:0 = ‘F0B’h, and read/write access is through register 15h.

The contents for this register are only valid for the PHY Address strapped by the PHYA[4:0] pins. For example if the PHYA[4:0] pins are strapped to 0001, then only Port 1’s Expansion Register 0Bh is valid. Bits[7:0] are latched high if the interrupt mask for the port is unmasked in register 1Bh and that interrupt has occurred. Bits[7:0] are cleared when the bits in register 1Ah are cleared.

Table 612: Expansion Register 0Bh: Port Interrupt Status Register (Address 15h)

Bit	Name	R/W	Description	Default
15:8	Reserved	RO	Ignore on read.	00h
7	Port 8 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 8. Cleared when Register 1Ah is read for Port 8.	0
6	Port 7 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 7. Cleared when Register 1Ah is read for Port 7.	0
5	Port 6 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 6. Cleared when Register 1Ah is read for Port 6.	0
4	Port 5 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 5. Cleared when Register 1Ah is read for Port 5.	0
3	Port 4 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 4. Cleared when Register 1Ah is read for Port 4.	0
2	Port 3 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 3. Cleared when Register 1Ah is read for Port 3.	0
1	Port 2 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 2. Cleared when Register 1Ah is read for Port 2.	0
0	Port 1 Interrupt Status	RO LH	0 = No interrupt has occurred since last read. 1 = An interrupt has occurred on Port 1. Cleared when Register 1Ah is read for Port 1.	0

EXPANSION REGISTER 10H: CABLE DIAGNOSTIC CONTROLS AND BASIC STATUS REGISTER (ADDRESS 15H)

Expansion register 10h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F10’h, and read/write access is through register 15h.

Table 613: Expansion Register 10h: Cable Diagnostic Controls and Basic Status Register (Address 15h)

Bit	Name	R/W	Description	Default
15:12	Channel Finished	RO	1 = Per pair, cable diagnostic is finished.	0000
11	Short Between Pairs Found	RO	1 = Detected at least one short between cable pairs. 0 = No shorts detected.	0
10	Open/Short on any Pair	RO	1 = Detected an open or short on at least one pair. 0 = No open or shorts detected.	0
9	All Finished	RO	1 = Cable diagnostic finished on all four pairs. 0 = Cable diagnostic not finished on all four pairs.	0
8	Enable Short Between Pair Detection	R/W	1 = Enable detection of shorts between cable pairs. 0 = Disable detection of shorts between cable pairs.	0
7	Two Pass	R/W	1 = Enable two-pass cable diagnostic algorithm.	1
6	Natural Link Cable Threshold Select	R/W	1 = Use regular cable-length thresholds for natural-link cable-length computation.	0
5	Reserved	R/W	Must be written as 0.	0
4	Reserved	R/W	Must be written as 0.	0
3	Phase Tune	R/W	1 = Enable phase-search algorithm—overrides Natural Link Control bit.	0
2	Natural Link	R/W	1 = Disable forced convergence of Echo taps. Also skip checking for open/shorts, and use special length-checking thresholds.	0
1	Cable Diagnostic Begin	R/W SC	1 = Begin cable-diagnostic algorithm.	0
0	Cable Diagnostic Mode	R/W	1 = Enable Cable-Diagnostic mode.	0

EXPANSION REGISTER 11H: CABLE DIAGNOSTIC RESULTS REGISTER (ADDRESS 15H)

Expansion register 11h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F11’h, and read/write access is through register 15h.

Table 614: Expansion Register 11h: Cable Diagnostic Results Register (Address 15h)

Bit	Name	R/W	Description	Default
15:12	Error	RO	1 = Per pair, error has occurred.	0000
11:8	Reserved	RO	Write as 0, ignore on read.	0000
7:6	Open/Short Pair 3	RO	Results of open/short detection on cable pair 3. 00 = Good cable. 01 = Cable open. 10 = Cable short. 11 = Cable broken (indeterminate).	00
5:4	Open/Short Pair 2	RO	Results of open/short detection on cable pair 2. 00 = Good cable. 01 = Cable open. 10 = Cable short. 11 = Cable broken (indeterminate).	0
3:2	Open/Short Pair 1	RO	Results of open/short detection on cable pair 1. 00 = Good cable. 01 = Cable open. 10 = Cable short. 11 = Cable broken (indeterminate).	0
1:0	Open/Short Pair 0	RO	Results of open/short detection on cable pair 0. 00 = Good cable. 01 = Cable open. 10 = Cable short. 11 = Cable broken (indeterminate).	0

EXPANSION REGISTER 12H: CABLE DIAGNOSTIC LENGTHS REGISTER (ADDRESS 15H)

Expansion register 12h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F12’h, and read/write access is through register 15h.

Table 615: Expansion Register 12h: Cable Diagnostic Lengths Register (Address 15h)

Bit	Name	R/W	Description	Default
15:8	Pair 1 Length	RO	Open/short length or cable length (meters).	00h
7:0	Pair 0 Length	RO	Open/short length or cable length (meters).	00h

EXPANSION REGISTER 13H: CABLE DIAGNOSTIC LENGTHS REGISTER (ADDRESS 15H)

Expansion register 13h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F13’h, and read/write access is through register 15h.

Table 616: Expansion Register 13h: Cable Diagnostic Lengths Register (Address 15h)

Bit	Name	R/W	Description	Default
15:8	Pair 3 Length	RO	Open/short length or cable length (meters).	00h
7:0	Pair 2 Length	RO	Open/short length or cable length (meters).	00h

EXPANSION REGISTER 42H: OPERATING MODE STATUS REGISTER (ADDRESS 15H)

Expansion register 42h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F42’h, and read/write access is through register 15h.

Table 617: Expansion Register 42h: Operating Mode Status Register (Address 15h)

Bit	Name	R/W	Description			Default
15	SerDes Link	RO	1 = Link up in SGMII, or Media Converter modes. 0 = Link down.			0
14	SerDes Speed	RO	Bit 14	Bit13	Speed	10
13			1	0	SGMII 1000	
			0	1	SGMII 100	
			0	0	SGMII 10	
12	SerDes Duplex	RO	1 = 1000BASE-X Full-Duplex. 0 = 1000BASE-X Half-Duplex.			0
11	Copper Link	RO	1 = Link up in 10BASE-T, 100BASE-TX, 1000BASE-T.			0
10	Copper Speed	RO	Bit 10	Bit 09	Speed	10
9			1	0	1000BASE-T	
			0	1	100BASE-TX	
			0	0	10BASE-T	
8	Copper Duplex	RO	1 = Copper Full-Duplex. 0 = Copper Half-Duplex.			0
7	Copper Energy Detect	RO	1 = Copper energy detected. 0 = No copper energy detected.			0
6	Fiber Signal Detect	RO	1 = Fiber signal detect from LED4 pin. 0 = No fiber signal detect from LED4 pin.			0
5	Sync Status	RO	1 = Valid SerDes PCS receive synchronization. 0 = No SerDes PCS receive synchronization.			0
4	Operating Mode Status	RO	Bits[4:0]	Operating Mode		See note.
3			10101	SGMII to Copper		
2			10111	Media Converter Mode		
1			Note: Default set by INTF_SEL pin.			
0						

EXPANSION REGISTER 44H: SGMII LINESIDE/LOOPBACK CONTROL REGISTER (ADDRESS 15H)

Expansion register 44h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F44’h, and read/write access is through register 15h.

Table 618: Expansion Register 44h: SGMII Lineside/Loopback Control Register (Address 15h)

Bit	Name	R/W	Description	Default
15:4	Reserved	R/W	Write as 0, ignore on read	000h
3	SGMII Lineside Loopback	R/W	1 = All data received on the MDI is looped back and transmitted on the MDI. If bit 0 = 0, then the data received on the MDI is also transmitted on the SGMII. 0 = Normal operation.	0
2	SGMII Loopback	R/W	1 = All data received on the SGMII is looped back and transmitted on the SGMII. If bit 0 = 0, then the data received on the MDI is also transmitted on the SGMII. If bit 1 = 0, then the data received on the SGMII is also transmitted on the MDI. 0 = Normal operation.	0
1	Transmit Suppress	R/W	1 = Suppress data received on the SGMII to MDI. 0 = Normal operation.	0
0	Receive Suppress	R/W	1 = Suppress data received on the MDI to SGMII. 0 = Normal operation.	0

EXPANSION REGISTER 51H: SGOUT± CONTROL REGISTER (ADDRESS 15H)

Expansion register 51h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F51’h, and read/write access is through register 15h.

Table 619: Expansion Register 51h: SGOUT± Control Register (Address 15h)

Bit	Name	R/W	Description	Default
15:12	Reserved	R/W	Write as 1001, ignore on read.	1001
11:8	Reserved	R/W	Write as 0111, ignore on read.	0111
7:4	Reserved	R/W	Write as 1001, ignore on read.	1001
3:0	Media Converter Transmit Amplitude	R/W	These four bits control the transit amplitude when in Media Converter mode. The approximate amplitude settings for the different settings are given below: 0000 = 0 mVppd 0001 = 100 mVppd 0010 = 200 mVppd 0011 = 300 mVppd 0100 = 400 mVppd 0101 = 500 mVppd 0110 = 600 mVppd 0111 = 700 mVppd 1000 = 800 mVppd 1001 = 900 mVppd 1010 = 1000 mVppd 1011 = 1100 mVppd 1100 = 1200 mVppd 1101 = 1300 mVppd (Default Value) 1110 = 1400 mVppd 1111 = 1500 mVppd	1101

EXPANSION REGISTER 52H: SGOUT \pm AND SCLK \pm CONTROL REGISTER (ADDRESS 15H)

Expansion register 52h is enabled by writing to “Expansion Register Access Register (Address 17h)” bits[11:0] = ‘F52’h, and read/write access is through register 15h.

Table 620: Expansion Register 52h: SGOUT \pm and SCLK \pm Control Register (Address 15h)

Bit	Name	R/W	Description	Default
15:8	Reserved	R/W	Write as 0, ignore on read.	00h
7:4	SGMII Transmit Amplitude	R/W	<p>These four bits control the transit amplitude when in SGMII mode. The approximate amplitude settings for the different settings are given below:</p> <p>0000 = 0 mVppd 0001 = 100 mVppd 0010 = 200 mVppd 0011 = 300 mVppd 0100 = 400 mVppd 0101 = 500 mVppd 0110 = 600 mVppd 0111 = 700 mVppd (Default Value) 1000 = 800 mVppd 1001 = 900 mVppd 1010 = 1000 mVppd 1011 = 1100 mVppd 1100 = 1200 mVppd 1101 = 1300 mVppd 1110 = 1400 mVppd 1111 = 1500 mVppd</p>	0111
3	Reserved	R/W	Write as 0, ignore on read.	0
2	SCLK \pm Disable	R/W	<p>1 = Disable SCLK\pm output. 0 = Enable SCLK\pm output.</p>	1
1	Reserved	R/W	Write as 0, ignore on read.	0
0	SGOUT \pm Disable	R/W	<p>1 = Disable SGOUT\pm output. 0 = Enable SGOUT\pm output.</p>	0

SCLK \pm Disable

Setting bit 2 = 1 disables the SCLK \pm output.

SGOUT \pm Disable

Setting bit 0 = 1 disables the SGOUT \pm output.

EXPANSION REGISTER 67H: 1000BASE-X AUTO-NEG Misc. RX STATUS REGISTER (ADDRESS 15H)

Expansion register 67h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F67’h, and read/write access is through register 15h.

Table 621: Expansion Register 67h: 1000BASE-X Auto-Neg Misc. RX Status Register (Address 15h)

Bit	Name	R/W	Description	Default
15	Auto-Negotiation Next Page Wait State Detected	RO LH	1 = Auto-negotiation Next Page Wait State entered since last read. 0 = Auto-negotiation Next Page Wait State not entered since last read.	0
14:0	Reserved	RO	Ignore on read.	0000h

Auto-Negotiation Next Page Wait State Detected

When bit 15 = 1, Auto-negotiation Next Page Wait State has been entered since last read.

EXPANSION REGISTER 70H: SOFT RESET REGISTER (ADDRESS 15H)

Expansion register 70h is enabled by writing to "Expansion Register Access Register (Address 17h)" bits[11:0] = 'F70'h, and read/write access is through register 15h.

Table 622: Expansion Register 70h: Soft Reset Register (Address 15h)

Bit	Name	R/W	Description	Default
15:3	Reserved	R/W	Write as 0, ignore on read.	0001h
2:1	Reserved	R/W	Write as 0, ignore on read.	00
0	Soft-Reset	R/W SC	1 = Issue soft-reset for 640 ns that will clear all registers in the NetXtreme II except for MDIO control registers. All MDIO status registers will be cleared. 0 = Normal operation.	0

Soft-Reset

Setting bit 0 = 1 will issue soft-reset for 640 ns that clears all registers in the NetXtreme II except for MDIO control registers. All MDIO status registers will be cleared.

EXPANSION REGISTER 71H: SERIAL LED CONTROL 1 REGISTER (ADDRESS 15H)

Expansion register 71h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F71’h, and read/write access is through register 15h.

Table 623: Expansion Register 71h: Serial LED Control 1 Register (Address 15h)

Bit	Name	R/W	Description	Default
15:14	Reserved	R/W	Write as 00, ignore on read.	00
13	Serial LED Enable	R/W	1 = Enable Serial LED mode. 0 = Disable Serial LED mode.	0
12	Low Cost Serial LED Enable	R/W	1 = Enable Low Cost Serial LED mode. 0 = Disable Low Cost Serial LED mode.	0
11:8	Serial LED bit 6	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u> 0010 = <u>XMITLED</u> 0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = Reserved 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0000
7:4	Serial LED bit 5	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u> 0010 = <u>XMITLED</u> 0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = Reserved 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0001

Table 623: Expansion Register 71h: Serial LED Control 1 Register (Address 15h)

Bit	Name	R/W	Description	Default
3:0	Serial LED bit 4	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u> 0010 = <u>XMITLED</u> 0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = Reserved 1011 = CABLE DIAGNOSTIC OPEN/SHORT 1100 = RESERVED 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0011

Serial LED Enable

Setting bit 13 = 1 will enable Serial LED mode.

Low Cost Serial LED Enable

Setting bit 12 = 1 will enable Low Cost Serial LED mode.

Serial LED Bit 6

Bits[11:8] can be used to program bit 6. The default for bit 6 is LINKSPD[1].

Serial LED Bit 5

Bits[7:4] can be used to program bit 5. The default for bit 5 is LINKSPD[2].

Serial LED Bit 4

Bits[3:0] can be used to program bit 4. The default for bit 4 is ACTIVITY.

EXPANSION REGISTER 72H: SERIAL LED CONTROL 2 REGISTER (ADDRESS 15H)

Expansion register 72h is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F72’h, and read/write access is through register 15h.

Table 624: Expansion Register 72h: Serial LED Control 2 Register (Address 15h)

Bit	Name	R/W	Description	Default
15:12	Reserved	R/W	Write as 00, ignore on read.	0h
11:8	Serial LED bit 3	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u> 0010 = <u>XMITLED</u> 0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = Reserved 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0100
7:4	Serial LED bit 2	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u> 0010 = <u>XMITLED</u> 0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = Reserved 1011 = <u>CABLE DIAGNOSTIC OPEN/SHORT</u> 1100 = <u>RESERVED</u> 1101 = <u>CRS</u> (SGMII mode) 1110 = OFF (output driven high) 1111 = ON (output driven low)	0110

Table 624: Expansion Register 72h: Serial LED Control 2 Register (Address 15h)

Bit	Name	R/W	Description	Default
3:0	Serial LED bit 1	R/W	0000 = <u>LINKSPD[1]</u> 0001 = <u>LINKSPD[2]</u> 0010 = <u>XMITLED</u> 0011 = <u>ACTIVITY</u> 0100 = <u>FDXLED</u> 0101 = <u>SLAVE</u> 0110 = <u>INTR</u> 0111 = <u>QUALITY</u> 1000 = <u>RCVLED</u> 1001 = <u>WIRESPD_DOWNGRADE</u> 1010 = Reserved 1011 = CABLE DIAGNOSTIC OPEN/SHORT 1100 = RESERVED 1101 = CRS (SGMII mode) 1110 = Off (output driven high) 1111 = On (output driven low)	1011

Serial LED Enable

Setting bit 13 = 1 will enable Serial LED mode.

Low Cost Serial LED Enable

Setting bit 12 = 1 will enable Low Cost Serial LED mode.

Serial LED Bit 3

Bits[11:8] can be used to program bit 3. The default for bit 3 is FDXLED.

Serial LED Bit 2

Bits[7:4] can be used to program bit 2. The default for bit 2 is INTR.

Serial LED Bit 1

Bits[3:0] can be used to program bit 1. The default for bit 1 is CABLE DIAGNOSTIC OPEN/SHORT.

EXPANSION REGISTER 7Bh: BSC CONTROL REGISTER (ADDRESS 15H)

Expansion register 7Bh is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F7B’h, and read/write access is through register 15h.

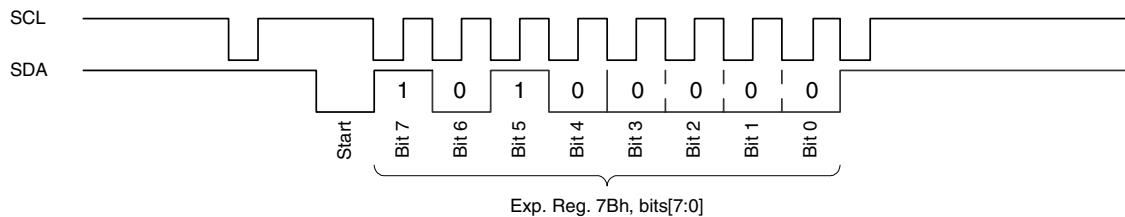
Table 625: Expansion Register 7Bh: BSC Control Register (Address 15h)

Bit	Name	R/W	Description	Default
15:12	BSC Command[3:0]	R/W	BSC Command[3:0] SC 0000 = IDLE (Read Only) 0001 = Initiate Write 0010 = Initiate Read 0100 = Initiate Stop 0110 = Load Registers from PROM 0111 = Write Byte	0000
11:8	Reserved	R/W	Write as 0, ignore on read.	0h
7:0	BSC Write Data	R/W	Data to be programmed to PROM.	0h

BSC Command[3:0]

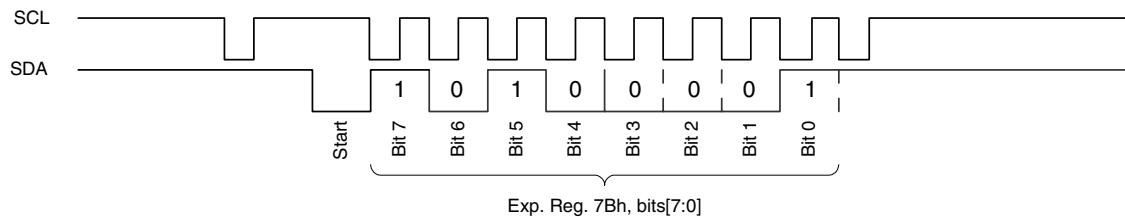
- The PHY will set bits[15:12] = 0000 (Idle state) after a BSC command has been accepted by the PHY's internal state machine.
- Setting bits[15:12] = 0001 and bits[7:0] will initiate a write command. Bits[7:0] are used to write the Device Code, Device Address, and Read/Write bit.

Initiate Write: Expansion Register 7Bh, bits [15:12] = 0001



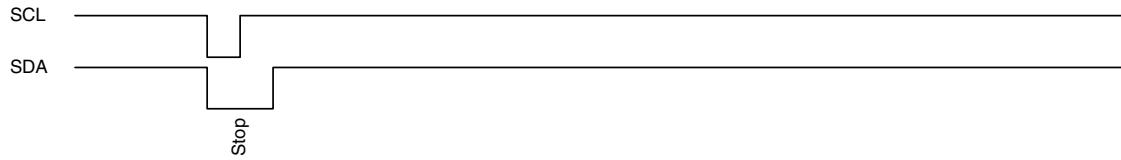
- Setting bits[15:12] = 0010 and bits[7:0] will initiate a read command. Bits[7:0] are used to write the Device Code, Device Address, and Read/Write bit.

Initiate Read: Expansion Register 7Bh, bits[15:12] = 0010



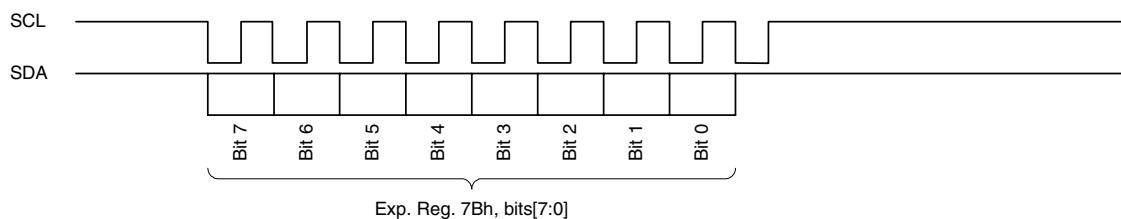
- Setting bits[15:12] = 0100 and bits[7:0] will initiate a stop command.

Initiate Stop: Expansion Register 7Bh, bits[15:12] = 0100



- Setting bits[15:12] = 0110 and bits[7:0] will cause the EEPROM to be reloaded.
- Setting bits[15:12] = 0111 and bits[7:0] will initiate a write byte command. Bits[7:0] are used to write either the word address or the data.

Write Byte: Expansion Register 7Bh, bits[15:12] = 0111



EXPANSION REGISTER 7Ch: BSC STATUS REGISTER (ADDRESS 15H)

Expansion register 7Ch is enabled by writing to “[Expansion Register Access Register \(Address 17h\)](#)” bits[11:0] = ‘F7C’h, and read/write access is through register 15h.

Table 626: Expansion Register 7Ch: BSC Status Register (Address 15h)

Bit	Name	R/W	Description	Default
15	PROM Not Present	RO SC	1 = PROM is not present. 0 = PROM present.	0
14	No ACK Received	RO SC	1 = No acknowledgement received after the determination that a PROM is present. 0 = Acknowledgement received.	0
13	Write Timer Expired	RO	1 = 5.2ms write timer has expired	0
12	PROM Size > 2048 bytes	RO	1 = PROM size > 2048 bytes. 0 = PROM size ≤ 2048 bytes.	
11	Unknown PROM Type	RO	1 = PHY did not see a 0x01 or a 0x02 programmed in the first byte of the PROM. 0 = Known PROM type.	0
10	Unprogrammed PROM	RO	1 = Did not see a x09 programmed in address 4 or a 0x13 programmed in address 5 of the PROM. 0 = PROM is programmed.	0
9	Programming Error	RO SC	1 = Programming error has occurred. 0 = No programming error has occurred.	0
8	Data Available	RO/SC	1 = Data available to read from bits[7:0]	0
7:0	Read Data	RO	Ignore on read	00h

PROM Not Present

Bit 15 is set = 1 when the NetXtreme II’s state machine issues the initial start followed by the device code and no acknowledgement is received from the PROM.

No ACK Received

Bit 14 is set = 1 when no acknowledgement is received during any BSC transaction after the determination that a PROM is connected. It indicates that there is some error in the transaction.

Write Timer Expired

The writing of a PROM’s internal buffer to its EE memory can take up to 5 ms. The PHY has a 5.2 ms timer that is reset after a write is executed. During this time, the PROM will not respond to its device code and address. The PHY will continuously transmit the device code and address and check for acknowledgement until it either receives it or the 5.2 ms timer expires. Once the timer expires the PHY will set bit 13 = 1 and transmit a Stop command

PROM Size > 2048 bytes

Bit 12 is set = 1 if during the memory determination sequence the NetXtreme II found that a PROM is larger than 2048 bytes. It is not cleared by the reading of the status register.

Unknown PROM Type

Bit 11 is set = 1 if neither a 0x01 or a 0x02 is the first byte read during the memory determination sequence.

Unprogrammed PROM

Bit 10 is set = 1 if word address 3 in the PROM does not contain a x09 or word address 4 in the PROM does not contain 0x13. This is a simple check to keep the state machine from loading large amounts of data into the NetXtreme II.

Programming Error

Bit 9 is set = 1 if a programming sequence error has occurred when an attempt is made to issue an MDIO command out of sequence in a normal BSC transaction.

Data Available

Bit 8 is set = 1 during a read sequence when valid data is present in bits[7:0].

Read Data

If bit 8 = 1, then bits[7:0] contain the data for the EEPROM's word address that was accessed during the read transaction. Each time these bits are read it will contain the data for the next word address as long as bit 8 = 1.



Broadcom Corporation

5300 California Avenue
Irvine, CA 92617
Phone: 949-926-5000
Fax: 949-926-5203

Broadcom® Corporation reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design.
Information furnished by Broadcom Corporation is believed to be accurate and reliable. However, Broadcom Corporation
does not assume any liability arising out of the application or use of this information, nor the application or use of any product or
circuit described herein, neither does it convey any license under its patent rights nor the rights of others.